MC9S08AC60
MC9S08AC48
MC9S08AC32

数据手册

HCS08
微处理器

MC9S08AC60ZHS
第 2 版
2008 年 3 月

freescale.com



MC9S08AC60 系列微控制器特性

8 位 HCS08 中央处理器 （CPU）

- 40 MHz HCS08 CPU (中央处理器)
- 20 MHz 内部总线频率 i
- HC08 指令集，带附加的 BGND 指令

开发支持

- 后台调试系统
- 断点功能，在在线调试 （在片上调试模块中另加两个断点）过程中允许进行单断点设置
- 片上在线仿真 (ICE) 调试模块包含两个比较器和九个触发模式。 8 个深度 FIFO，存储流更改地址及纯事件数据。支持标签和强制断点。
- 支持最多 32 个中断 / 复位源

存储器

- 最多 60 KB 片上 FLASH 存储器，带安全选项
- 最多 2 KB 片上 RAM

时钟源

- 可选时钟源包括晶振、谐振器、外部时钟或内部生成的时钟，通过 ICG 模块进行精确度 NVM 调整

系统保护

- 可选监视微控制器正常操作的看门狗 (COP) 复位，可以选择独立的 1kHz 内部时钟源或总线时钟
- 复位或中断的低压检测
- 复位非法操作码检测
- 循环冗余校验 （CRC）模块，支持对存储器的快速循环冗余校验

省电模式

- 等待模式加两个停止模式

外围设备

- **ADC** — 最多 16 路、 10 位模数转换器，带自动比较功能
- **SCI** — 两个串行通信接口模块，带可选 13 位中断。支持 LIN 2.0 协议和 SAE J2602 ；主扩展中断生成；从扩展中断检测
- **SPI** — 串行外围接口模块
- **IIC** — 内部集成电路总线模块支持最高 100 kbps 总线负载，如果降低负载能够支持更高的波特率。 10 位地址扩展选项
- **定时器** — 最多两个 2 路和一个 6 路 16 位定时器 / 脉宽调制器 (TPM) 模块：每个通路上可以选择输入捕获，输出对比和边沿对齐 PWM 功能。每个定时器模块可以配置为所有通路都设置缓冲、中心对齐 PWM (CPWM) 。
- **KBI** — 最多 8 引脚键盘中断模块
- **CRC** — 采用 16 位移位寄存器产生硬件 CRC

输入 / 输出

- 最多 54 个通用输入 / 输出 (I/O) 引脚
- 当用作输入时，端口上拉器件软件可选
- 当用作输出时，端口斜率控制软件可选
- 当用作输出时，端口驱动强度软件可选
- 主复位引脚和加电复位 (POR)
- RESET、 IRQ 和 BKGD/MS 引脚上的内部上拉器件，降低客户系统成本

封装

- 64 引脚四方扁平封装 (QFP)
- 64 引脚小尺寸四方扁平封装 (LQFP)
- 48 引脚四方扁平无引线封装 (QFN)
- 44 引脚小尺寸四方扁平封装 (LQFP)
- 32 引脚小尺寸四方扁平封装 (LQFP)

MC9S08AC60 系列 数据手册

包括　　MC9S08AC60
MC9S08AC48
MC9S08AC32

MC9S08AC60 系列
第 2 版
2008 年 3 月



修改历史

我们在网上提供这些文档的最新版本。印刷文档可能是之前的版本。请访问以下网站，确认您获得了最新信息：

http://freescale.com/

下表列出了对本文进行的修改。为了方便起见，指示的页码已经链接到相应的位置。

修订版本编号	修订日期	更改说明
1	2/2008	初始版本
2	3/2008	市场推出版本

章节号	标题	页

手册目录

章节号 **标题** **页码**

第 1 章
介绍

第 2 章
引脚和连接

第 3 章
操作模式

第 4 章
存储器

第 5 章
复位、中断和系统配置

第 6 章
并行输入 / 输出

第 7 章
中央处理单元 （S08CPUV2）

第 8 章
循环冗余校验 (S08CRCV1)

第 9 章
模数转换器 (S08ADC10V1)

第 10 章
内部时钟发生器 (S08ICGV4)

第 11 章
IIC 模块 (S08IICV2)

第 12 章
键盘中断 (S08KBIV1)

第 13 章
串行通信接口 (S08SCIV4)

第 14 章
串行外设接口 (S08SPIV3)

第 15 章
定时器 /PWM (S08TPMV3)

第 16 章
开发支持

附录 A
电气特性和时序规范

附录 B
订购信息和机械图

第 1 章
介绍

1.1 概述

MC9S08AC60 系列 MCU 是低成本、高性能 HCS08 系列 8 位微处理器单元 （MCU）的成员。这个系列的所有 MCU 都采用增强型 HCS08 内核，提供各种模块、存储器大小、存储器类型及封装类型。表 1-1 列出了存储器大小及封装类型。

表 1-1. MC9S08AC60 系列中包含的器件

器件	FLASH	RAM	封装
MC9S08AC60	63,280	2048	64 QFP 64 LQFP 48 QFN 44 LQFP 32 LQFP
MC9S08AC48	49,152		
MC9S08AC32	32,768		

表 1-2 总结了 MC9S08AC60 系列 MCU 的特性。

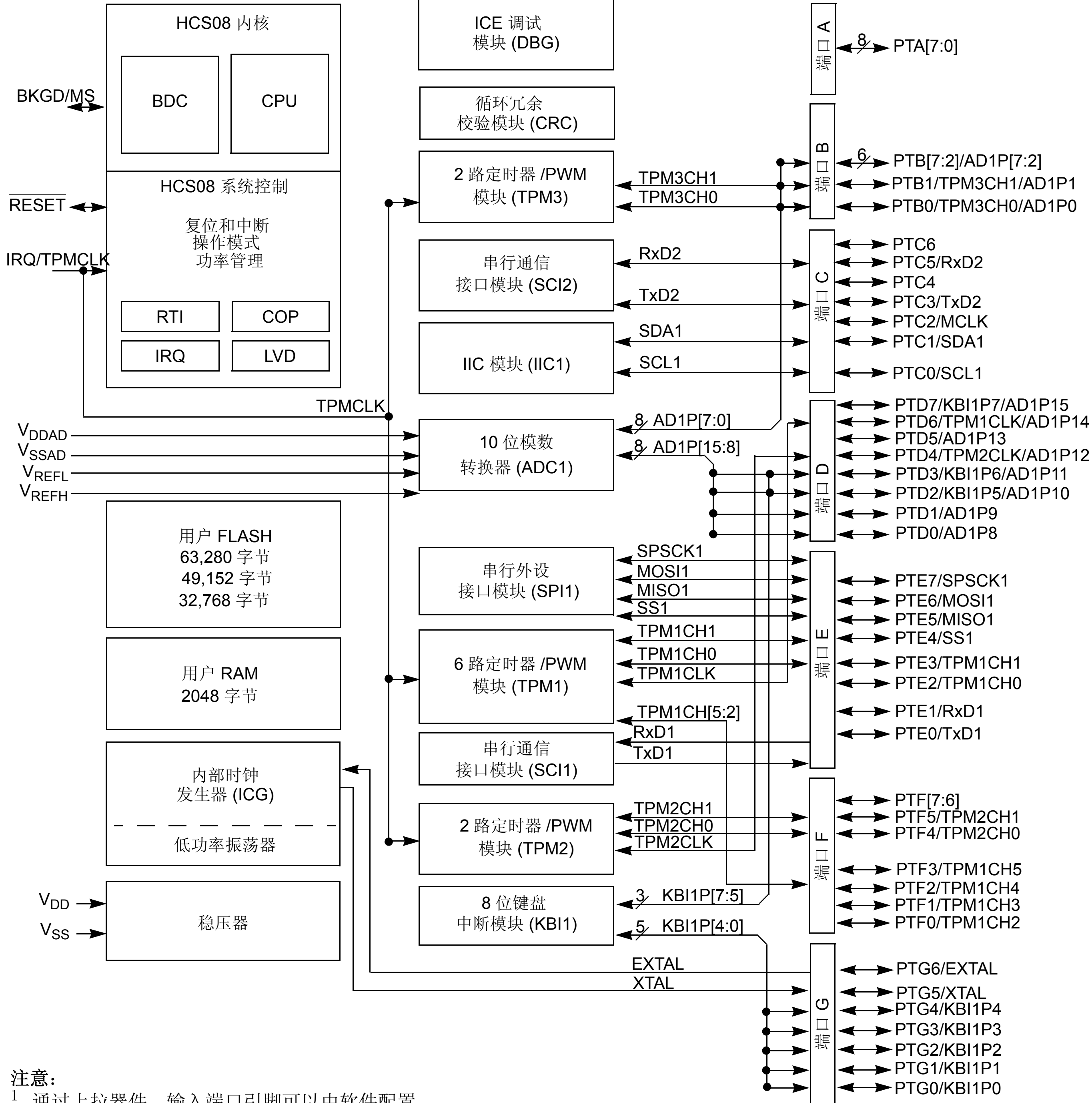
表 1-2. MC9S08AC60 系列每种封装提供的外设

<table>
<tr><th></th><th colspan="4">MC9S08AC60/48/32</th></tr>
<tr><th>特性</th><th>64 引脚</th><th>48 引脚</th><th>44 引脚</th><th>32 引脚</th></tr>
<tr><td>CRC</td><td colspan="4">有</td></tr>
<tr><td>ADC</td><td>16 路</td><td colspan="2">8 路</td><td>6 路</td></tr>
<tr><td>IIC</td><td colspan="4">有</td></tr>
<tr><td>IRQ</td><td colspan="4">有</td></tr>
<tr><td>KBI1</td><td>8</td><td>7</td><td>6</td><td>4</td></tr>
<tr><td>SCI1</td><td colspan="4">有</td></tr>
<tr><td>SCI2</td><td colspan="3">有</td><td>no</td></tr>
<tr><td>SPI1</td><td colspan="4">有</td></tr>
<tr><td>TPM1</td><td>6 路</td><td colspan="2">4 路</td><td>2 路</td></tr>
<tr><td>TPM1CLK[1]</td><td>有</td><td colspan="3">无</td></tr>
<tr><td>TPM2</td><td colspan="4">2 路</td></tr>
<tr><td>TPM2CLK[1]</td><td>有</td><td colspan="3">无</td></tr>
<tr><td>TPM3</td><td colspan="4">2 路</td></tr>
<tr><td>TPMCLK [1]</td><td colspan="4">有</td></tr>
<tr><td>I/O 引脚</td><td>54</td><td>38</td><td>34</td><td>22</td></tr>
</table>

[1] TPMCLK、 TPM1CLK 和 TPM2CLK 选项可通过软件配置； 复位后， TPM1CLK、 TPM2CLK 和 TPMCLK 分别提供给 TPM1、 TPM2 和 TPM3。请参阅 TPM 章节，了解 TPMxCLK 信号的功能描述。

1.2 MCU 结构图

下面为 MC9S08AC60 系列 MCU 的模块结构图。



注意：

[1] 通过上拉器件，输入端口引脚可以由软件配置。

[2] 如果 IRQ 使能 (IRQPE = 1)，引脚包含软件可配置的上拉 / 下拉器件。如果选择上升边沿检测，则下拉使能 (IRQEDG = 1)。

[3] 引脚包含集成上拉器件。

[4] PTD3、PTD2、PTD7 和 PTG4 既包含上拉也包含下拉器件。当 KBI 使能时 (KBIPEn = 1)，且选择了上升边沿 (KBEDGn = 1)，则下拉使能 。

[5] TPMCLK、 TPM1CLK 和 TPM2CLK 选项可通过软件配置； 复位后， TPM1CLK、 TPM2CLK 和 TPMCLK 分别提供给 TPM1、 TPM2 和 TPM3。

图 1-1. MC9S08AC60 系列结构图

表 1-3 列出了片上模块的版本。 .

表 1-3. . 片上模块的版本

模块		版本
循环冗余校验发生器	CRC	1
模数转换器	ADC	1
内部时钟发生器	ICG	4
内置集成电路	IIC	2
键盘中断	KBI	1
串行通信接口	SCI	4
串行外设接口	SPI	3
定时器和脉宽调制器	TPM	3
中央处理单元	CPU	2
调试模块	DBG	2

1.3 系统时钟分配

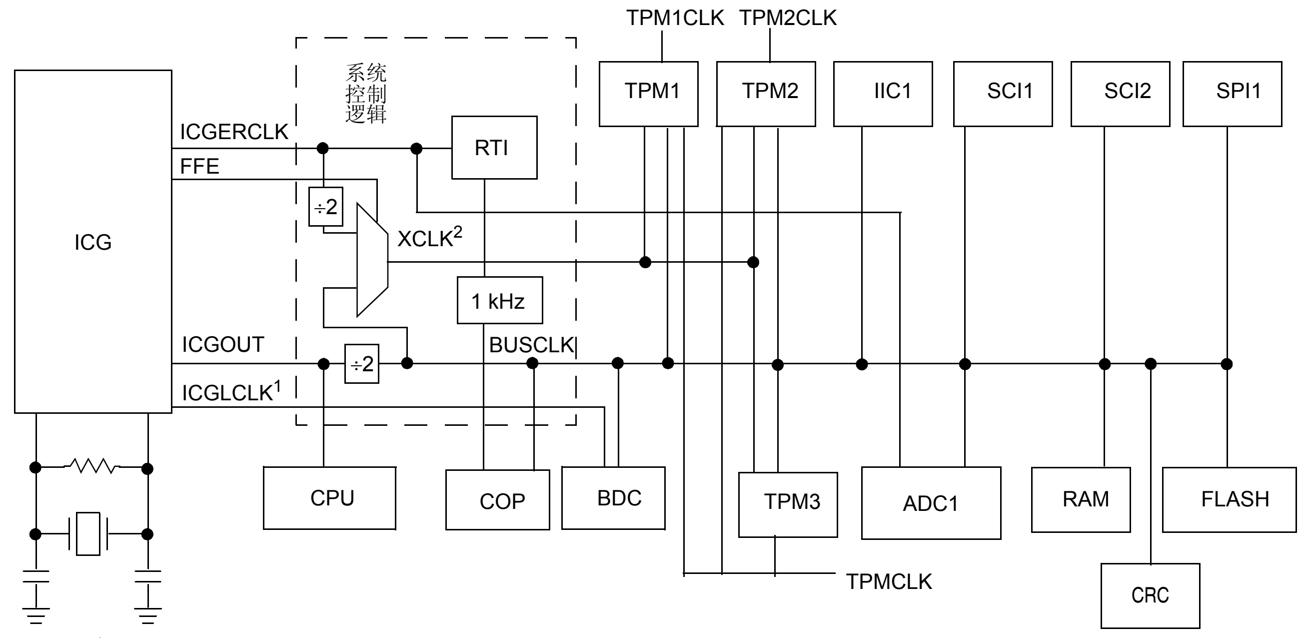


图 1-2. 系统时钟分配图

该 MCU 中的某些模块具有多个时钟源可选择。图 1-2 是一个简化的时钟连接图。 ICG 提供以下时钟源：

- ICGOUT 是 ICG 模块的输出。可以是下面任一项：
  - 外部晶体振荡器
  - 外部时钟源
  - 锁频环子模块中数控振荡器 （DCO）的输出

  - ICG 中的控制位决定连接哪一个时钟源。
- FFE 是 ICG 中生成的控制信号。如果 ICGOUT 的频率 > 4 × ICGERCLK 的频率，该信号为逻辑 1，固定频率时钟将为 ICGERCLK/2。否则，固定频率时钟将为 BUSCLK。
- ICGLCLK — 开发工具可以选择这个内部自时钟源 (~ 8 MHz) 来加快系统 （总线时钟比较慢）中的 BDC 通信。
- ICGERCLK — 外部参考时钟可以作为实时中断时钟源。它也可以用作 ADC 模块的 ALTCLK 输入。

第 2 章
引脚和连接

2.1 介绍

本章描述了连接到封装引脚的信号，其中包括引脚布局图、信号属性表，并对信号进行了详细阐述。

2.2 器件引脚分配

图 2-1 为 MC9S08AC60 系列的 64 引脚封装图。

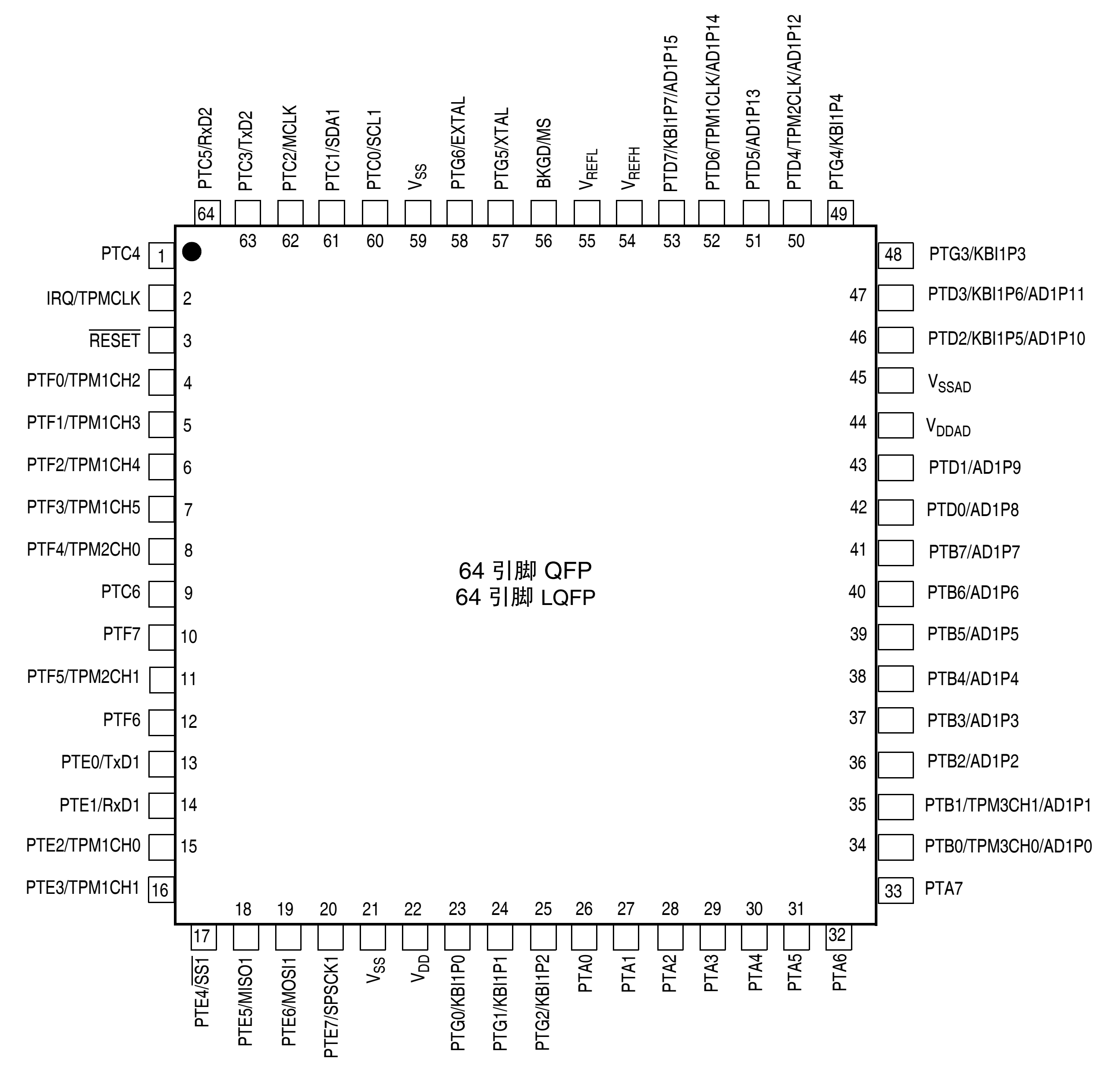


图 2-1. MC9S08AC60 系列 64 引脚 QFP 和 LQFP 封装

图 2-2 为 MC9S08AC60 系列的 48 引脚 QFN 封装图。

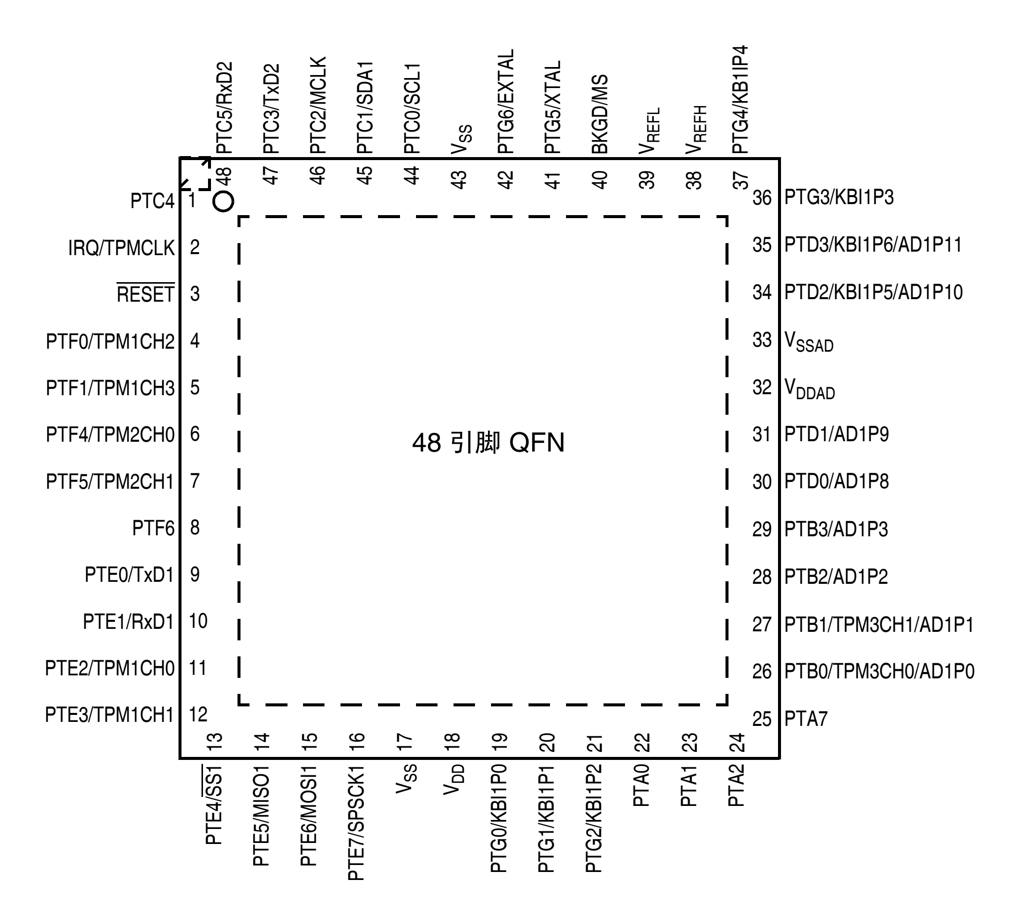


图 2-2. MC9S08AC60 系列 48 引脚 QFN 封装

图 2-3 为 MC9S08AC60 系列微控制器 44 引脚 LQFP 封装图。

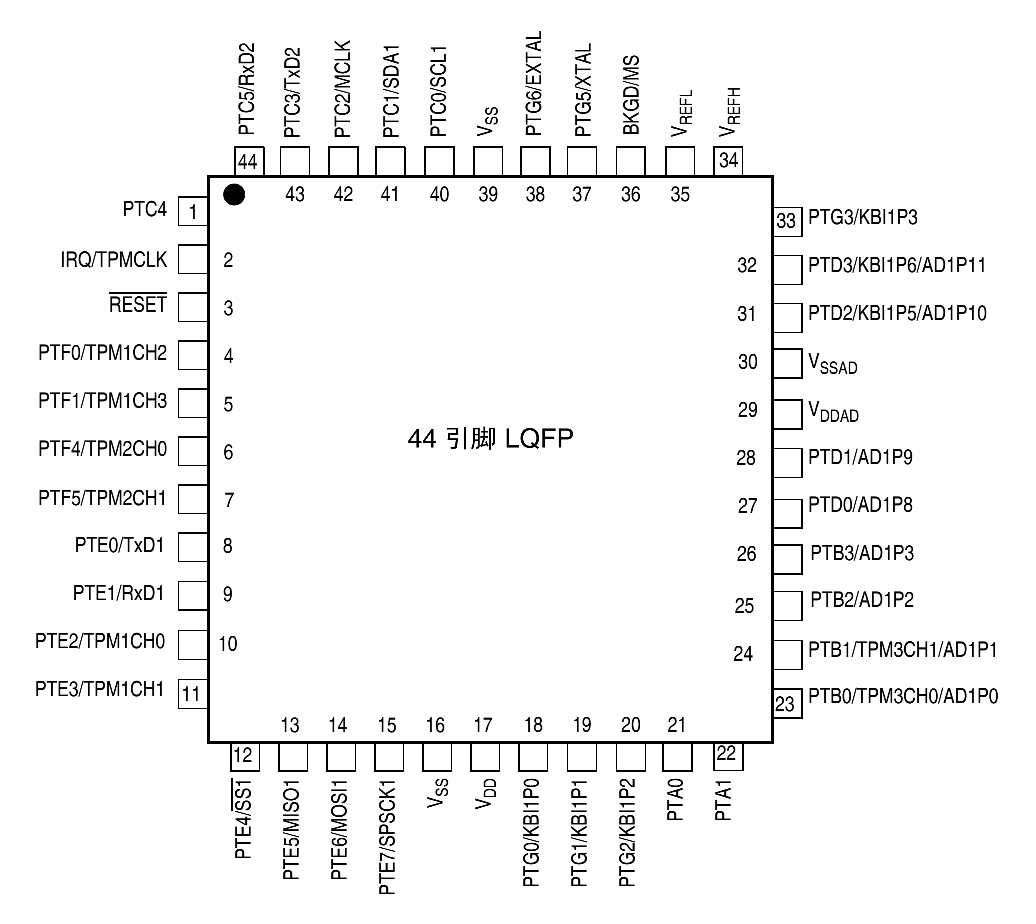


图 2-3. MC9S08AC60 系列 44 引脚 LQFP 封装

图 2-4 为 MC9S08AC60 系列微控制器 32 引脚 LQFP 封装图。

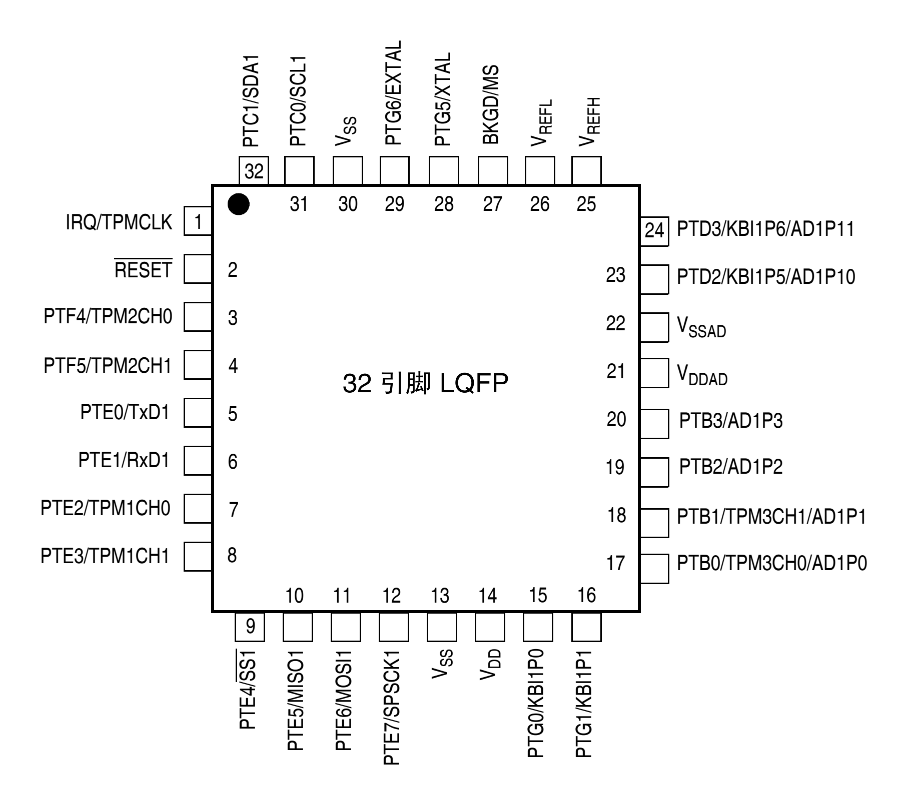
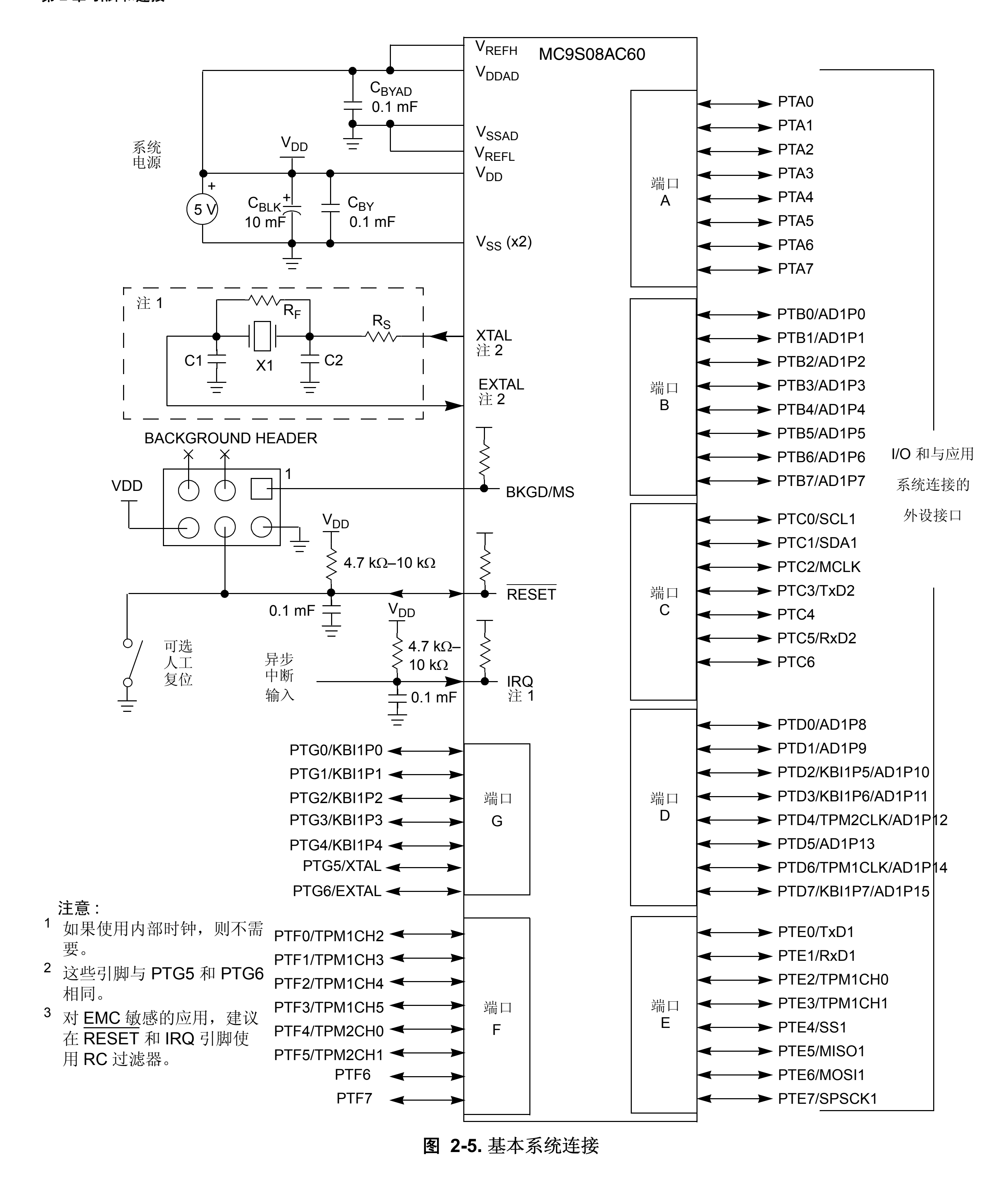


图 2-4. MC9S08AC60 系列 32 引脚 LQFP 封装

2.3 推荐的系统连接

图 2-5 给出了 MC9S08AC60 系列应用系统的通用引脚连接。



MC9S08AC60
VREFH
VDDAD
CBYAD
0.1 mF
VSSAD
VREFL
VDD
系统
电源
5 V
CBLK
10 mF
CBY
0.1 mF
VSS (x2)
注 1
RF
RS
C1
X1
C2
XTAL
注 2
EXTAL
注 2
BACKGROUND HEADER
VDD
BKGD/MS
4.7 kΩ–10 kΩ
0.1 mF
RESET
可选
人工
复位
异步
中断
输入
4.7 kΩ–
10 kΩ
0.1 mF
IRQ
注 1
端口 A
PTA0
PTA1
PTA2
PTA3
PTA4
PTA5
PTA6
PTA7
端口 B
PTB0/AD1P0
PTB1/AD1P1
PTB2/AD1P2
PTB3/AD1P3
PTB4/AD1P4
PTB5/AD1P5
PTB6/AD1P6
PTB7/AD1P7
I/O 和与应用
系统连接的
外设接口
端口 C
PTC0/SCL1
PTC1/SDA1
PTC2/MCLK
PTC3/TxD2
PTC4
PTC5/RxD2
PTC6
端口 D
PTD0/AD1P8
PTD1/AD1P9
PTD2/KBI1P5/AD1P10
PTD3/KBI1P6/AD1P11
PTD4/TPM2CLK/AD1P12
PTD5/AD1P13
PTD6/TPM1CLK/AD1P14
PTD7/KBI1P7/AD1P15
端口 E
PTE0/TxD1
PTE1/RxD1
PTE2/TPM1CH0
PTE3/TPM1CH1
PTE4/SS1
PTE5/MISO1
PTE6/MOSI1
PTE7/SPSCK1
端口 G
PTG0/KBI1P0
PTG1/KBI1P1
PTG2/KBI1P2
PTG3/KBI1P3
PTG4/KBI1P4
PTG5/XTAL
PTG6/EXTAL
端口 F
PTF0/TPM1CH2
PTF1/TPM1CH3
PTF2/TPM1CH4
PTF3/TPM1CH5
PTF4/TPM2CH0
PTF5/TPM2CH1
PTF6
PTF7
注意：
1 如果使用内部时钟，则不需要。
2 这些引脚与 PTG5 和 PTG6 相同。
3 对 EMC 敏感的应用，建议在 RESET 和 IRQ 引脚使用 RC 过滤器。

图 2-5. 基本系统连接

2.3.1 电源 (V_{DD}, V_{SS}, V_{DDAD}, V_{SSAD})

V_{DD} 和 V_{SS} 是 MCU 的主电源引脚。该电压源为所有 I/O 缓冲电路以及内部稳压器供电。内部稳压器向 CPU 和 MCU 的其它内部电路提供稳压后的低电压源。

通常，应用系统在电源引脚上需要安装两个独立的电容器，其中一个为大容量电解电容器 （如 10 µF 钽电容器），为整个系统提供大容量电荷存储，同时应在离 V_{DD} 和 V_{SS} 电源引脚尽可能近的地方安装一个 0.1 µF 陶瓷旁路电容器，来抑制高频噪音。MC9S08AC60 还多一个 V_{SS} 引脚。这个引脚 应该连接到系统接地平面或者通过一个低阻抗连接到主 V_{SS} 引脚 。

V_{DDAD} 和 V_{SSAD} 是 MCU 的模拟电源引脚。该电源为 ADC 模块供电。必须在离该模拟电源引脚尽可能近的地方连接一个 0.1 µF 陶瓷旁路电容，来抑制高频噪音。

2.3.2 振荡器 (XTAL, EXTAL)

复位后，MCU 使用内部生成的时钟（自时钟模式 — f_{Self_reset}），相当于 8 MHz 晶体振荡频率。这个频率的时钟源在复位启动时使用，可以作为停止恢复的时钟源，这样可以避免较长的晶体启动延迟。该 MCU 还包含一个可修整的内部时钟发生器 (ICG) 模块，可以用它来运行 MCU。有关 ICG 的更多信息，请参见 第 10 章 内部时钟发生器 (S08ICGV4)。

MCU 中的振荡器是 Pierce 振荡器，通过 ICGC1 寄存器 RANGE 位设置，可以提供两个频率范围给 晶体或陶瓷谐振器选择。 EXTAL 输入引脚连接的是一个外部振荡器，而不是晶体或陶瓷谐振器。

下面的讨论参考图 2-5 。R_S (如果使用) 和 R_F 应该采用无感电阻器，例如碳复合材料的电阻器 。线绕电阻和一些金属膜电阻器的电感太大。 C1 和 C2 应该采用高质量的陶瓷电容器，专用于高频。

R_F 用来提供一个旁路，使 EXTAL 输入在晶振启动时保持在其线性范围内，它的值一般来说不太关键。典型系统使用 1 MΩ 至 10 MΩ 范围内的 R_F 。值太高，对湿度就敏感，值太低，会减少增益，（在极端情况下）会阻止启动。

C1 和 C2 一般在 5 pF 至 25 pF 范围内，要选择符合特定晶体或陶瓷谐振器要求的值。在确定 C1 和 C2 值时，一定要考虑印刷电路板 （PCB）电容和 MCU 引脚电容。晶振生产商一般规定一个负载电容，它是 C1 和 C2 的系列组合，C1 和 C2 的大小通常相同。首次逼近时，采用 10 pF 作为引脚 和 PCB 组合电容的估算值接入每个振荡器引脚 (EXTAL 和 XTAL)。

2.3.3 $\overline{\text{RESET}}$ 引脚

$\overline{\text{RESET}}$ 是一个专用的引脚，内置一个上拉器件。它有输入滞后，包含一个高电流输出驱动，无输出斜率控制。内部加电复位和低压复压电路一般都不需要外部复位电路。这个引脚通常连接到标准的 6 引脚后台调试连接器，这样开发系统可以直接复位 MCU 系统。如果需要，可以添加一个人工外部复位，即提供一个简单的接地开关 （拉低复位引脚的电平，强制进行复位）。

只要发起任何复位 (无论是来自外部信号还是来自内部系统)，复位引脚的电平就下降 34 个周期的 $f_{自复位}$。复位电路解断复位的原因并将之记录在系统控制复位状态寄存器 （SRS）的相应位。

对 EMC 敏感的应用，建议在复位引脚上加一个外部 RC 过滤器。参见 图 2-5 的示例。

2.3.4 后台 / 模式选择 (BKGD/MS)

在复位时， BKGD/MS 引脚充当模式选择引脚。复位完成后，该引脚立即作为后台引脚 ，可以用于后台调试通信。当作为后台 / 模式模式选择引脚时，该引脚包括一个内部上拉器件，有输入滞后，且无输出斜率控

制。当引脚作为 后台引脚时，它包括一个高电流输出驱动。当该引脚作为模式选择引脚时，它只有输入，因此不包含标准的输出驱动。

如果该引脚没有连接，MCU 将在复位的上升边沿进入正常的操作模式 。如果有一个调试系统连接到这个 6 引脚标准后台调试头，它可以在复用上升边沿使保持 BKGD/MS 在低电平，强制 MCU 进入激活后台模式。

BKGD 引脚主要用于后台调试控制器 (BDC) 通信，采用定制的协议。这个协议每位时采用 16 个周期的目标 MCU 的 BDC 时钟 。目标 MCU 的 BDC 时钟可以与总线时钟速率一样快，这样就不会有任何大电容连接到 BKGD/MS 引脚，来干扰后台串行通信。

尽管 BKGD 引脚 是一个伪开路引脚，后台调试通信协议提供短暂的、主动驱动的高速脉冲，确保快速的上升时间。电缆的小电容和内部上拉器件的绝对值几乎不影响 BKGD 引脚的上升和下降时间。

2.3.5 ADC 参考引脚 (V_{REFH}, V_{REFL})

V_{REFH} 和 V_{REFL} 引脚分别为 ADC 模块的电压参考高输入和电压参考低输入。

2.3.6 外部中断引脚 (IRQ)

IRQ 引脚是 IRQ 中断的输入源，也是 BIH 和 BIL 指令的输入。如果未使能 IRQ 功能，这个引脚仍配置为 TPMCLK (参见 TPM 章节)。

对 EMC 敏感的应用中，建议在 IRQ 引脚上添加一个外部 RC 过滤器。参见 图 2-5 的示例。

2.3.7 通用 I/O 和外设端口

剩余的引脚被通用 I/O 和片上外设功能，如定时器和串行 I/O 系统共用。复位后，所有这些引脚立即配置为高阻抗通用输入，且内部上拉器件关闭。

注意

为了避免浮动输入引脚消耗额外的电流，应用程序中的复位初始化例程必须使能片上上拉器件或者更改未使用的引脚到输出的方向，使引脚不浮空。

不是所有封装都提供所有的通用 I/O 引脚。为了避免浮空输入引脚消耗额外的电流，应用程序中用户的复位初始化例程必须使能片上上拉器件或者更改未使用的引脚到输出的方向，使引脚不浮动。

有关控制这些引脚作为通用 I/O 引脚的更多信息，参见第 6 章 并行输入 / 输出。

当片上外围系统控制引脚时，数据方向控制位仍决定从端口数据寄存器中读什么，而外围模块则通过控制引脚输出缓冲的使能来控制引脚方向。参见第 6 章 并行输入 / 输出 了解更详细的信息。

只要引脚作为输入，每个输入引脚的上拉使能位控制片上上拉器件是否使能，即使引脚是受片上外围模块控制。当 PTD7、PTD3、PTD2 和 PTG4 引脚由 KBI 模块控制，且配置为上升边沿 / 高电平敏感时，上拉使能控制位使能下拉器件，而不是上拉器件。同样，当 IRQ 配置为 IRQ 输入，且设置为检测上升边沿时，上拉使能控制位使能下拉器件，而不是上拉器件。

注意

如果先激活替代功能，则可能会给模块一个伪边沿。用户软件必须在激活中断前清除所有相关的标记。表 2-1 列出了多个模块使能时的优先级。最高优先级模块对引脚有控制权。在使能了较低优先级的功能后，再选择较高优先级引脚功能，会造成较低优先级模块得到伪边沿。建议在启动另一个模块前关闭所有共享此引脚的模块。

表 2-1. 引脚提供情况 （按封装引脚计数）

引脚编号				<-- 最低 优先级 --> 最高		
64	**48**	**44**	**32**	**端口引脚**	**Alt 1**	**Alt 2**
1	1	1	—	PTC4		
2	2	2	1	IRQ	TPMCLK[1]	
3	3	3	2	RESET		
4	4	4	—	PTF0	TPM1CH2	
5	5	5	—	PTF1	TPM1CH3	
6	—	—	—	PTF2	TPM1CH4	
7	—	—	—	PTF3	TPM1CH5	
8	6	6	3	PTF4	TPM2CH0	
9	—	—	—	PTC6		
10	—	—	—	PTF7		
11	7	7	4	PTF5	TPM2CH1	
12	8	—	—	PTF6		
13	9	8	5	PTE0	TxD1	
14	10	9	6	PTE1	RxD1	
15	11	10	7	PTE2	TPM1CH0	
16	12	11	8	PTE3	TPM1CH1	
17	13	12	9	PTE4	SS1	
18	14	13	10	PTE5	MISO1	
19	15	14	11	PTE6	MOSI1	
20	16	15	12	PTE7	SPSCK1	
21	17	16	13	V_{SS}		
22	18	17	14	V_{DD}		
23	19	18	15	PTG0	KBI1P0	
24	20	19	16	PTG1	KBI1P1	
25	21	20	—	PTG2	KBI1P2	
26	22	21	—	PTA0		
27	23	22	—	PTA1		
28	24	—	—	PTA2		
29	—	—	—	PTA3		
30	—	—	—	PTA4		
31	—	—	—	PTA5		
32	—	—	—	PTA6		
33	25	—	—	PTA7		
34	26	23	17	PTB0	TPM3CH0	AD1P0
35	27	24	18	PTB1	TPM3CH1	AD1P1
36	28	25	19	PTB2	AD1P2	
37	29	26	20	PTB3	AD1P3	
38	—	—	—	PTB4	AD1P4	
39	—	—	—	PTB5	AD1P5	
40	—	—	—	PTB6	AD1P6	
41	—	—	—	PTB7	AD1P7	
42	30	27	—	PTD0	AD1P8	
43	31	28	—	PTD1	AD1P9	
44	32	29	21	V_{DDAD}		
45	33	30	22	V_{SSAD}		
46	34	31	23	PTD2	KBI1P5	AD1P10
47	35	32	24	PTD3	KBI1P6	AD1P11
48	36	33	—	PTG3	KBI1P3	
49	37	—	—	PTG4	KBI1P4	
50	—	—	—	PTD4	TPM2CLK	AD1P12
51	—	—	—	PTD5	AD1P13	
52	—	—	—	PTD6	TPM1CLK	AD1P14
53	—	—	—	PTD7	KBI1P7	AD1P15
54	38	34	25	V_{REFH}		
55	39	35	26	V_{REFL}		
56	40	36	27	BKGD	MS	
57	41	37	28	PTG5	XTAL	
58	42	38	29	PTG6	EXTAL	
59	43	39	30	V_{SS}		
60	44	40	31	PTC0	SCL1	
61	45	41	32	PTC1	SDA1	
62	46	42	—	PTC2	MCLK	
63	47	43	—	PTC3	TxD2	
64	48	44	—	PTC5	RxD2	

1. TPMCLK、 TPM1CLK 和 TPM2CLK 可通过软件配置； 复位后， TPM1CLK、 TPM2CLK 和 TPMCLK 分别提供给 TPM1、 TPM2 和 TPM3。

第 3 章
操作模式

3.1 介绍

本章介绍了 MC9S08AC60 系列的操作模式，

3.2 特性

- 用于代码开发的激活后台模式
- 等待模式：
  - CPU 关闭以省电
  - 系统时钟正常运行
  - 稳压功能正常运行
- 停止模式：
  - 系统时钟停止；稳压器处于待机状态
  - 停止 2：内部电路的部分电源关闭； 保留 RAM 内容
  - 停止 3：所有内部电路都接通电源，以进行快速恢复

3.3 运行模式

这是 MC9S08AC60 系列。如果 BKGD/MS 引脚处于复位上升边沿的高电平位置，则进入这个模式。在这种模式下，CPU 从内部存储器执行代码，从地址 0xFFFE: 0xFFFF 开始。

3.4 激活后台模式

激活后台模式功能通过 HCS08 内核中的后台调试控制器 （BDC）进行管理。在软件开发过程中，BDC 与片上 ICE 调试模块 （DBG）一起用于分析 MCU 的 操作。

通过以下五种方式可以进入后台模式：

- BKGD/MS 引脚在上升边沿的低电平时
- 通过 BKGD 引脚收到 BACKGROUND 指令时
- 执行 BGND 指令时
- 遇到 BDC 断点时
- 遇到 DBG 断点时

进入激活后台模式后，CPU 保持挂起状态，等待串行后台指令而不是执行来自用户应用程序的指令。

后台指令有两种类型：

- 非侵入式指令：用户程序运行时可以发出的指令，在 MCU 处于运行模式时通过 BKGD 引脚发出。它们也可以在 MCU 处于激活后台模式时执行。非入侵式指令包括：
  - 存储器访问指令

    - 带状态的存储器访问指令
    - BDC 寄存器访问指令
    - BACKGROUND 指令
- 激活后台指令：只有当 MCU 处于激活后台模式时才执行的指令。激活后台指令包含执行以下操作的命令：
    - 读或写 CPU 寄存器
    - 在特定时间跟踪一个用户程序指令
    - 退出激活后台模式，返回用户应用程序 （GO）

3.5 等待模式

执行 WAIT 指令可以进入等待模式。在执行 WAIT 指令后，CPU 进入无时钟的低功耗状态。 CPU 进入等待模式后， CCR （条件代码寄存器）中的 I 位被清除，启动中断操作。发生中断请求后， CPU 退出等待模式并恢复处理，先开始执行堆叠操作，进入中断服务例程。

MCU 处于等待模式时，后台调试命令的使用受到一些限制。 MCU 处于等待模式时，只有 BACKGROUND 指令和带状态存储器访问指令可用。带状态存储器访问指令虽然禁止存储器访问，但它们会上报错误，指示 MCU 处于停止或等待模式。可以使用后台命令将 MCU 从等待模式中唤醒进入激活后台模式。

3.6 停止模式

当系统选项寄存器中设置了 STOPE 位，在执行 STOP 指令时，进入停止模式 （共两个停止模式）。在停止模式中，所有内部时钟都暂停。如果在 CPU 执行 STOP 指令时， STOPE 位没有设置，则 MCU 不能进入任何一个停止模式，并且强制执行非法操作代码复位。在 SPMSC2 中设置相应位可以选择停止模式。

一些用于低电压 (1.8 V 至 3.6 V) 操作的 HCS08 器件还包含停止 1 模式。 MC9S08AC60 系列 的器件在 2.7 V 至 5.5 V 范围内操作，不包含停止 1 模式。

表 3-1 总结了每个停止模式中 MCU 的行为。

表 3-1. 停止模式行为

模式	PPDC	CPU, 数字外设，FLASH	RAM	ICG	ADC	稳压器	I/O 引脚	RTI
停止 2	1	关闭	待机	关闭	关闭	待机	状态保持	可选打开
停止 3	0	待机	待机	关闭[1]	可选打开	待机	状态保持	可选打开

[1] 晶体振荡器可以设置在停止 3 模式中运行。参见 ICG 寄存器。

3.6.1 停止 2 模式

停止 2 模式的待机功耗非常低，而且它保持 RAM 的内容以及所有 I/O 引脚的电流状态。要进入停止 2 模式，用户必须执行 STOP 指令，选择停止 2 (PPDC = 1) 的，使能停止模式 (STOPE = 1)。此外，在停止模式操作时，LVD 必须关闭 (LVDSE = LVDE = 1)。如果在停止模式中 LVD 使能，则在执行 STOP 指令时 MCU 进入停止 3 模式，无论 PPDC 处于何状态。

在进入停止 2 模式之 前，用户必须将 I/O 端口寄存器的内容，以及任何其它存储器映射的寄存器的内容（如果需要在退出停止 2 模式后恢复），保存到 RAM 中的位置。在退出停止 2 模式时，这些值可以在引脚锁存打开之前通过用户软件恢复。

当 MCU 处于停止 2 模式时，除 RAM 外，所有通过稳压器供电的内部电路都被关闭。稳压器与 ADC 一样处于低功率的待机状态。在进入停止 2 模式时， I/O 引脚的状态被锁存。状态在停止 2 模式中和退出停止 2 模式后保持不变，直到写 1 到 SPMSC2 中的 PPDACK。

通过确认唤醒引脚：$\overline{\text{RESET}}$ 或 IRQ，或通过 RTI 中断可以退出停止 2 模式。MCU 处于停止 2 模式时，IRQ 总是激活低输入，无论进入停止 2 模式之前它是如何设置的。

从停止 2 模式唤醒时，除了 引脚状态仍被锁存， MCU 的启动与加电复位 (POR) 相似。CPU 将取复位向量。系统及所有外设将处于其默认复位状态，必须初始化。

从停止 2 模式唤醒后，SPMSC2 中的 PPDF 位被设置。这个标记可以用来引导用户代码运行停止 2 恢复例程。PPDF 保留设置，I/O 引脚状态仍被锁存，直到写逻辑 1 到 SPMSC2 中的 PPDACK。

要保持 （已配置为通用 I/O 的）引脚的 I/O 状态，在写入 PPDACK 位之前，用户必须将 I/O 端口寄存器的内容 （已经保存在 RAM 中）恢复到端口寄存器。如果在写入 PPDACK 之前，端口 寄存器没有从 RAM 恢复，则在 I/O 引脚锁存打开时，寄存器位将回到其复位状态，I/O 引脚将切换到其复位状态。

对于配置为外围 I/O 的引脚，在写入 PPDACK 位之前，用户必须重新配置与该引脚接口的外围模块。如果在写入 PPDACK 位之前，外围模块未使能，则当 I/O 锁存打开时，引脚将由其相关的端口控制寄存器控制。

实时中断采用单独的自时钟源 (≈1 kHz) ，允许从停止 2 或停止 3 唤醒，无需借助外部组件。当 RTIS2:RTIS1:RTIS0 = 0:0:0 时，则关闭实时中断功能和这个 1 kHz 的时钟源。当该时钟源关闭时，功耗更低，但在这种情况下，实时中断无法将 MCU 从停止模式唤醒。

3.6.2 停止 3 模式

要进入停止 3 模式，用户必须执行 STOP 指令，选择停止 3 模式 (PPDC = 0)，使能停止模式 (STOPE = 1)。在进入停止 3 模式时， MCU 中的所有时钟，包括振荡器本身都暂停。 ICG 与稳压器和 ADC 一样进入待机状态。所有内部寄存器和逻辑的状态，以及 RAM 内容均保持不变。和停止 2 模式不同， I/O 引脚状态不锁存于引脚，相反，它们的状态跟随驱动引脚保持状态的内部逻辑的状态而定。

通过确认 $\overline{\text{RESET}}$，或异步中断引脚，或通过实时中断 （RTI）可以退出停止 3 模式。异步中断引脚为 IRQ 或 KBI 引脚。通过 SCI 接收器中断， ADC 和 LVI 也可以退出停止 3 模式。

如果通过确认 $\overline{\text{RESET}}$ 引脚退出停止 3 模式， MCU 将复位，获取复位向量后，操作将恢复。如果通过异步中断或者实时中断退出， MCU 将获取相应的中断向量。

实时中断采用单独的自时钟源 (≈1 kHz) ，允许从停止 2 或停止 3 唤醒，无需借助外部组件。当 RTIS2:RTIS1:RTIS0 = 0:0:0 时，则关闭实时中断功能和这个 1 kHz 的时钟源。当 1 kHz 的时钟源关闭时，功耗更低，但在这种情况下，实时中断无法将 MCU 从停止模式唤醒。

3.6.3 在停止模式中使能激活 BDM

如果设置了 BDCSCR 中的 ENBDM 位，将启用从运行模式进入激活后台模式的操作。第 16 章 开发支持中描述了该寄存器。如果 CPU 执行 STOP 指令时，设置 ENBDM 位，则当 MCU 进入停止模式时，用于后台

调试逻辑的系统时钟仍保持激活状态。这样，仍然可以进行后台调试通信。此外，稳压器不进入低功率待机状态，而是保持正常工作。如果用户想通过设置 ENBDM 进入停止 2 模式，则 MCU 将进入停止 3 模式。

停止模式下大多数后台指令不可用。带状态存储器访问指令禁止存储器访问，但它们会上报错误，指示 MCU 处于停止或等待模式。如果 ENBDM 位已设置，可以使用 BACKGROUND 命令将 MCU 从停止模式中唤醒进入激活后台模式。进入后台调试模式后，所有后台命令都可以使用。表 3-2 总结了使能 “进入后台调试模式” 时 MCU 的行动。

表 3-2. BDM 使能的停止模式行为

模式	PPDC	CPU，数字外设，FLASH	RAM	ICG	ADC	稳压器	I/O 引脚	RTI
停止 3	x	待机	待机	激活	可选打开	激活	保持状态	可选打开

3.6.4 在停止模式中使能 LVD

在电源电压下降到 LVD 电压以下时，LVD 系统可以生成中断或复位。在 CPU 执行 STOP 指令时，如果 LVD 在停止模式下使能 （通过设置 SPMSC1 中的 LVDE 和 LVDSE 位），那么稳压器在停止模式下将继续保持激活状态。如果用户想通过启动 LVD 进入停止 2 模式，则 MCU 将进入停止 3 模式。表 3-3 总结了 LVD 使能时， MCU 在停止模式的行为。

表 3-3. LVD Enabled Stop Mode Behavior

模式	PPDC	CPU, Digital Peripherals, FLASH	RAM	ICG	ADC	稳压器	I/O 引脚	RTI
停止 3	x	待机	待机	关闭	可选打开	激活	保持状态	可选打开

3.6.5 停止模式下的片上外围模块

当 MCU 进入停止模式时，内部外围模块的系统时钟停止运行。即使在异常情况下 （ENBDM = 1），后台调试逻辑的时钟继续工作，外围系统的时钟也暂停以降低功耗。有关停止模式下系统行为的详细信息，请参见 3.6.1 节 停止 2 模式 和 3.6.2 节 停止 3 模式。

表 3-4. 停止模式行为

外设	模式	
	停止 2	停止 3
CPU	关闭	待机
RAM	待机	待机
FLASH	关闭	待机
并行端口寄存器	关闭	待机
ADC	关闭	可选打开[1]
ICG	关闭	可选打开[2]
IIC	关闭	待机
RTI	可选打开[3]	可选打开 [3]
SCI	关闭	待机

表 3-4. 停止模式行为 (续)

外设	模式	
	停止 2	停止 3
SPI	关闭	待机
TPM	关闭	待机
系统稳压器	待机	待机
I/O 引脚	状态保持	状态保持

1 要求启动异步 ADC 时钟和 LVD，否则为待机。

2 ICGC1 中的 OSCSTEN 被设置，否则为待机。

3 进入停止模式前， SRTISC 中的 RTIS[2:0] 不为 0，否则关闭。

第 4 章
存储器

4.1 MC9S08AC60 系列

如 图 4-1 所示，MC9S08AC60 系列 MCU 中的片上存储器包括 RAM、非易失性数据存储的 Flash 存储器、 I/O 和控制 / 状态寄存器。这些寄存器可分为以下三类：

- 直接页面寄存器 ($0000 至 $006F)
- 高位页面寄存器 ($1800 至 $185F)
- 非易失性寄存器 ($FFB0 至 $FFBF)

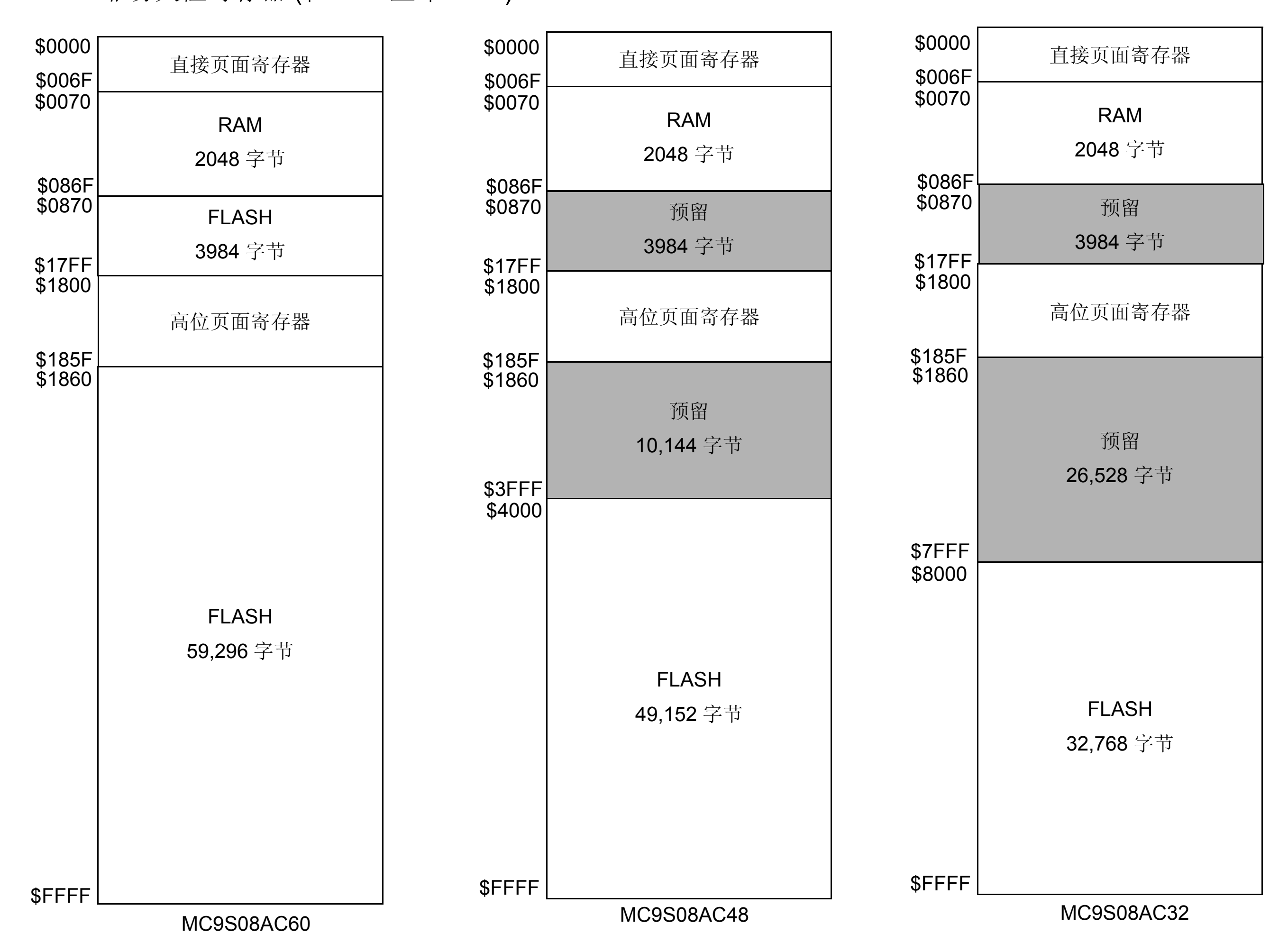


图 4-1. MC9S08AC60 系列存储器映射

4.1.1 复位和中断向量分配

图 4-1 为复位和中断向量的地址分配情况。该表中使用的向量名称为飞思卡尔半导体提供的 MC9S08AC60 系列同等文件中使用的标签。有关复位、中断、中断优先级和本地中断掩码控制的更多信息，请参见 第 5 章 复位、中断和系统配置。

表 4-1. 复位和中断向量

地址 (高 / 低)	向量	向量名称
0xFFC0:FFC1 到 0xFFC4:FFC5	未使用的向量空间 (提供给用户程序)	—
0xFFC6:FFC7	TPM3 溢出	Vtpm3ovf
0xFFC8:FFC9	TPM3 通路 1	Vtpm3ch1
0xFFCA:FFCB	TPM3 通路 0	Vtpm3ch0
0xFFCC:FFCD	RTI	Vrti
0xFFCE:FFCF	IIC1	Viic1
0xFFD0:FFD1	ADC1 转换	Vadc1
0xFFD2:FFD3	KBI1	Vkeyboard1
0xFFD4:FFD5	SCI2 发送	Vsci2tx
0xFFD6:FFD7	SCI2 接收	Vsci2rx
0xFFD8:FFD9	SCI2 错误	Vsci2err
0xFFDA:FFDB	SCI1 发送	Vsci1tx
0xFFDC:FFDD	SCI1 接收	Vsci1rx
0xFFDE:FFDF	SCI1 错误	Vsci1err
0xFFE0:FFE1	SPI1	Vspi1
0xFFE2:FFE3	TPM2 溢出	Vtpm2ovf
0xFFE4:FFE5	TPM2 通路 1	Vtpm2ch1
0xFFE6:FFE7	TPM2 通路 0	Vtpm2ch0
0xFFE8:FFE9	TPM1 溢出	Vtpm1ovf
0xFFEA:FFEB	TPM1 通路 5	Vtpm1ch5
0xFFEC:FFED	TPM1 通路 4	Vtpm1ch4
0xFFEE:FFEF	TPM1 通路 3	Vtpm1ch3
0xFFF0:FFF1	TPM1 通路 2	Vtpm1ch2
0xFFF2:FFF3	TPM1 通路 1	Vtpm1ch1
0xFFF4:FFF5	TPM1 通路 0	Vtpm1ch0
0xFFF6:FFF7	ICG	Vicg
0xFFF8:FFF9	低电压检测	Vlvd
0xFFFA:FFFB	IRQ	Virq
0xFFFC:FFFD	SWI	Vswi
0xFFFE:FFFF	复位	Vreset

4.2 寄存器地址和位分配

MC9S08AC60 系列产品中的寄存器分为以下三个组：

- 直接页面寄存器，位于存储器映射的前 112 个位置上。因此通过高效的直接寻址模式指令就可以访问它们。
- 高位页面寄存器，不经常使用，因此在存储器映象中位于 0x1800 以上，为直接页面中经常使用的寄存器和变量留出了更多空间。
- 非易失性寄存器区域包含 16 个位置，位于 FLASH 中的 $FFB0–$FFBF。

  非易失性寄存器位置包含：

  - 三个值，在复位时加载到工作寄存器
  - 一个 8 字节后门比较密钥，可选地允许用户接入安全的存储器

  由于非易失性寄存器位置为 FLASH 存储器，它们必须象其它 FLASH 存储器一样被擦除和编程。

直接页面寄存器可以通过高效的直接寻址模式指令访问。可以通过位操作指令访问直接页面寄存器中的任何位。表 4-2 总结了所有用户可访问的直接页面寄存器和控制位。

表 4-2 中的直接页面寄存器可以用更高效的直接寻址模式（这种模式只需要较低字节）。因此，第一栏中地址的较低字节用粗体显示。表 4-3 和表 4-4 整个第一栏都用粗体显示。表 4-2、表 4-3 和表 4-4 中，第二栏的寄存器名称都用粗体显示，以区别右边的位名称。与所列出的位不相关的单元格用阴影显示。带 0 的阴影单元格表示这个未使用的位始终读为 0。带破折号的阴影单元格指示未使用的或预留的位可以读为 1 或 0。

表 4-2. 直接页面寄存器一览 (第 1 页，共 3 页)

地址	寄存器名称	Bit 7	6	5	4	3	2	1	Bit 0
0x0000	**PTAD**	PTAD7	PTAD6	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
0x0001	**PTADD**	PTADD7	PTADD6	PTADD5	PTADD4	PTADD3	PTADD2	PTADD1	PTADD0
0x0002	**PTBD**	PTBD7	PTBD6	PTBD5	PTBD4	PTBD3	PTBD2	PTBD1	PTBD0
0x0003	**PTBDD**	PTBDD7	PTBDD6	PTBDD5	PTBDD4	PTBDD3	PTBDD2	PTBDD1	PTBDD0
0x0004	**PTCD**	0	PTCD6	PTCD5	PTCD4	PTCD3	PTCD2	PTCD1	PTCD0
0x0005	**PTCDD**	0	PTCDD6	PTCDD5	PTCDD4	PTCDD3	PTCDD2	PTCDD1	PTCDD0
0x0006	**PTDD**	PTDD7	PTDD6	PTDD5	PTDD4	PTDD3	PTDD2	PTDD1	PTDD0
0x0007	**PTDDD**	PTDDD7	PTDDD6	PTDDD5	PTDDD4	PTDDD3	PTDDD2	PTDDD1	PTDDD0
0x0008	**PTED**	PTED7	PTED6	PTED5	PTED4	PTED3	PTED2	PTED1	PTED0
0x0009	**PTEDD**	PTEDD7	PTEDD6	PTEDD5	PTEDD4	PTEDD3	PTEDD2	PTEDD1	PTEDD0
0x000A	**PTFD**	PTFD7	PTFD6	PTFD5	PTFD4	PTFD3	PTFD2	PTFD1	PTFD0
0x000B	**PTFDD**	PTFDD7	PTFDD6	PTFDD5	PTFDD4	PTFDD3	PTFDD2	PTFDD1	PTFDD0
0x000C	**PTGD**	0	PTGD6	PTGD5	PTGD4	PTGD3	PTGD2	PTGD1	PTGD0
0x000D	**PTGDD**	0	PTGDD6	PTGDD5	PTGDD4	PTGDD3	PTGDD2	PTGDD1	PTGDD0
0x000E– 0x000F	预留	— —	— —	— —	— —	— —	— —	— —	— —
0x0010	**ADC1SC1**	COCO	AIEN	ADCO	ADCH				
0x0011	**ADC1SC2**	ADACT	ADTRG	ACFE	ACFGT	0	0	R	R
0x0012	**ADC1RH**	0	0	0	0	0	0	ADR9	ADR8
0x0013	**ADC1RL**	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
0x0014	**ADC1CVH**	0	0	0	0	0	0	ADCV9	ADCV8
0x0015	**ADC1CVL**	ADCV7	ADCV6	ADCV5	ADCV4	ADCV3	ADCV2	ADCV1	ADCV0
0x0016	**ADC1CFG**	ADLPC	ADIV		ADLSMP	MODE		ADICLK	
0x0017	**APCTL1**	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
0x0018	**APCTL2**	ADPC15	ADPC14	ADPC13	ADPC12	ADPC11	ADPC10	ADPC9	ADPC8
0x0019– 0x001B	预留	— —	— —	— —	— —	— —	— —	— —	— —
0x001C	**IRQSC**	0	IRQPDD	IRQEDG	IRQPE	IRQF	IRQACK	IRQIE	IRQMOD
0x001D	预留	—	—	—	—	—	—	—	—
0x001E	**KBISC**	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBF	KBACK	KBIE	KBIMOD
0x001F	**KBIPE**	KBIPE7	KBIPE6	KBIPE5	KBIPE4	KBIPE3	KBIPE2	KBIPE1	KBIPE0
0x0020	**TPM1SC**	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
0x0021	**TPM1CNTH**	Bit 15	14	13	12	11	10	9	Bit 8
0x0022	**TPM1CNTL**	Bit 7	6	5	4	3	2	1	Bit 0
0x0023	**TPM1MODH**	Bit 15	14	13	12	11	10	9	Bit 8
0x0024	**TPM1MODL**	Bit 7	6	5	4	3	2	1	Bit 0
0x0025	**TPM1C0SC**	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
0x0026	**TPM1C0VH**	Bit 15	14	13	12	11	10	9	Bit 8
0x0027	**TPM1C0VL**	Bit 7	6	5	4	3	2	1	Bit 0
0x0028	**TPM1C1SC**	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0

表 4-2. 直接页面寄存器一览 (第 2 页，共 3 页)

地址	寄存器名称	Bit 7	6	5	4	3	2	1	Bit 0
0x0029	TPM1C1VH	Bit 15	14	13	12	11	10	9	Bit 8
0x002A	TPM1C1VL	Bit 7	6	5	4	3	2	1	Bit 0
0x002B	TPM1C2SC	CH2F	CH2IE	MS2B	MS2A	ELS2B	ELS2A	0	0
0x002C	TPM1C2VH	Bit 15	14	13	12	11	10	9	Bit 8
0x002D	TPM1C2VL	Bit 7	6	5	4	3	2	1	Bit 0
0x002E	TPM1C3SC	CH3F	CH3IE	MS3B	MS3A	ELS3B	ELS3A	0	0
0x002F	TPM1C3VH	Bit 15	14	13	12	11	10	9	Bit 8
0x0030	TPM1C3VL	Bit 7	6	5	4	3	2	1	Bit 0
0x0031	TPM1C4SC	CH4F	CH4IE	MS4B	MS4A	ELS4B	ELS4A	0	0
0x0032	TPM1C4VH	Bit 15	14	13	12	11	10	9	Bit 8
0x0033	TPM1C4VL	Bit 7	6	5	4	3	2	1	Bit 0
0x0034	TPM1C5SC	CH3F	CH5IE	MS5B	MS5A	ELS5B	ELS5A	0	0
0x0035	TPM1C5VH	Bit 15	14	13	12	11	10	9	Bit 8
0x0036	TPM1C5VL	Bit 7	6	5	4	3	2	1	Bit 0
0x0037	预留	—	—	—	—	—	—	—	—
0x0038	SCI1BDH	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
0x0039	SCI1BDL	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
0x003A	SCI1C1	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
0x003B	SCI1C2	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
0x003C	SCI1S1	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
0x003D	SCI1S2	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
0x003E	SCI1C3	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
0x003F	SCI1D	Bit 7	6	5	4	3	2	1	Bit 0
0x0040	SCI2BDH	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
0x0041	SCI2BDL	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
0x0042	SCI2C1	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
0x0043	SCI2C2	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
0x0044	SCI2S1	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
0x0045	SCI2S2	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
0x0046	SCI2C3	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
0x0047	SCI2D	Bit 7	6	5	4	3	2	1	Bit 0
0x0048	ICGC1	HGO	RANGE	REFS	CLKS		OSCSTEN	LOCD	0
0x0049	ICGC2	LOLRE	MFD			LOCRE	RFD		
0x004A	ICGS1	CLKST		REFST	LOLS	LOCK	LOCS	ERCS	ICGIF
0x004B	ICGS2	0	0	0	0	0	0	0	DCOS
0x004C	ICGFLTU	0	0	0	0	FLT			
0x004D	ICGFLTL	FLT							
0x004E	ICGTRM	TRIM							
0x004F	预留	—	—	—	—	—	—	—	—
0x0050	SPI1C1	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE

表 4-2. 直接页面寄存器一览 (第 3 页，共 3 页)

地址	寄存器名称	Bit 7	6	5	4	3	2	1	Bit 0
0x0051	**SPI1C2**	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
0x0052	**SPI1BR**	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
0x0053	**SPI1S**	SPRF	0	SPTEF	MODF	0	0	0	0
0x0054	预留	0	0	0	0	0	0	0	0
0x0055	**SPI1D**	Bit 7	6	5	4	3	2	1	Bit 0
0x0057	**CRCL**	Bit 7	6	5	4	3	2	1	Bit 0
0x0058	**IIC1A**	ADDR							0
0x0059	**IIC1F**	MULT		ICR					
0x005A	**IIC1C1**	IICEN	IICIE	MST	TX	TXAK	RSTA	0	0
0x005B	**IIC1S**	TCF	IAAS	BUSY	ARBL	0	SRW	IICIF	RXAK
0x005C	**IIC1D**	DATA							
0x005D	**IIC1C2**	GCAEN	ADEXT	0	0	0	AD10	AD9	AD8
0x005E– 0x005F	预留	— —	— —	— —	— —	— —	— —	— —	— —
0x0060	**TPM2SC**	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
0x0061	**TPM2CNTH**	Bit 15	14	13	12	11	10	9	Bit 8
0x0062	**TPM2CNTL**	Bit 7	6	5	4	3	2	1	Bit 0
0x0063	**TPM2MODH**	Bit 15	14	13	12	11	10	9	Bit 8
0x0064	**TPM2MODL**	Bit 7	6	5	4	3	2	1	Bit 0
0x0065	**TPM2C0SC**	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
0x0066	**TPM2C0VH**	Bit 15	14	13	12	11	10	9	Bit 8
0x0067	**TPM2C0VL**	Bit 7	6	5	4	3	2	1	Bit 0
0x0068	**TPM2C1SC**	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
0x0069	**TPM2C1VH**	Bit 15	14	13	12	11	10	9	Bit 8
0x006A	**TPM2C1VL**	Bit 7	6	5	4	3	2	1	Bit 0
0x006B– 0x006F	预留	— —	— —	— —	— —	— —	— —	— —	— —

表 4-3 中显示的高位页面寄存器的访问频率比 I/O 和控制寄存器的低的多，因此，它们位于直接可寻址的存储器空间之外，从 0x1800 开始。

表 4-3. 高位页面寄存器一览 (第 1 页，共 2 页)

地址	寄存器名称	Bit 7	6	5	4	3	2	1	Bit 0
0x1800	SRS	POR	PIN	COP	ILOP	0	ICG	LVD	0
0x1801	SBDFR	0	0	0	0	0	0	0	BDFR
0x1802	SOPT	COPE	COPT	STOPE	—	0	0	—	—
0x1803	SMCLK	0	0	0	MPE	0	MCSEL		
0x1804 – 0x1805	预留	— —	— —	— —	— —	— —	— —	— —	— —
0x1806	SDIDH	REV3	REV2	REV1	REV0	ID11	ID10	ID9	ID8
0x1807	SDIDL	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
0x1808	SRTISC	RTIF	RTIACK	RTICLKS	RTIE	0	RTIS2	RTIS1	RTIS0
0x1809	SPMSC1	LVDF	LVDACK	LVDIE	LVDRE	LVDSE	LVDE	0[1]	BGBE
0x180A	SPMSC2	LVWF	LVWACK	LVDV	LVWV	PPDF	PPDACK	—	PPDC
0x180B	预留	—	—	—	—	—	—	—	—
0x180C	SOPT2	COPCLKS	—	—	—	TPMCCFG	—	—	—
0x180D– 0x180F	预留	— —	— —	— —	— —	— —	— —	— —	— —
0x1810	DBGCAH	Bit 15	14	13	12	11	10	9	Bit 8
0x1811	DBGCAL	Bit 7	6	5	4	3	2	1	Bit 0
0x1812	DBGCBH	Bit 15	14	13	12	11	10	9	Bit 8
0x1813	DBGCBL	Bit 7	6	5	4	3	2	1	Bit 0
0x1814	DBGFH	Bit 15	14	13	12	11	10	9	Bit 8
0x1815	DBGFL	Bit 7	6	5	4	3	2	1	Bit 0
0x1816	DBGC	DBGEN	ARM	TAG	BRKEN	RWA	RWAEN	RWB	RWBEN
0x1817	DBGT	TRGSEL	BEGIN	0	0	TRG3	TRG2	TRG1	TRG0
0x1818	DBGS	AF	BF	ARMF	0	CNT3	CNT2	CNT1	CNT0
0x1819– 0x181F	预留	— —	— —	— —	— —	— —	— —	— —	— —
0x1820	FCDIV	DIVLD	PRDIV8	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
0x1821	FOPT	KEYEN	FNORED	0	0	0	0	SEC01	SEC00
0x1822	预留	—	—	—	—	—	—	—	—
0x1823	FCNFG	0	0	KEYACC	0	0	0	0	0
0x1824	FPROT	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FPDIS
0x1825	FSTAT	FCBEF	FCCF	FPVIOL	FACCERR	0	FBLANK	0	0
0x1826	FCMD	FCMD7	FCMD6	FCMD5	FCMD4	FCMD3	FCMD2	FCMD1	FCMD0
0x1827– 0x182F	预留	— —	— —	— —	— —	— —	— —	— —	— —
0x1830	TPM3SC	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
0x1831	TPM3CNTH	Bit 15	14	13	12	11	10	9	Bit 8
0x1832	TPM3CNTL	Bit 7	6	5	4	3	2	1	Bit 0
0x1833	TPM3MODH	Bit 15	14	13	12	11	10	9	Bit 8
0x1834	TPM3MODL	Bit 7	6	5	4	3	2	1	Bit 0

表 4-3. 高位页面寄存器一览 (第 2 页，共 2 页)

地址	寄存器名称	Bit 7	6	5	4	3	2	1	Bit 0
0x1835	TPM3C0SC	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	0	0
0x1836	TPM3C0VH	Bit 15	14	13	12	11	10	9	Bit 8
0x1837	TPM3C0VL	Bit 7	6	5	4	3	2	1	Bit 0
0x1838	TPM3C1SC	CH1F	CH1IE	MS1B	MS1A	ELS1B	ELS1A	0	0
0x1839	TPM3C1VH	Bit 15	14	13	12	11	10	9	Bit 8
0x183A	TPM3C1VL	Bit 7	6	5	4	3	2	1	Bit 0
0x183B 0x183F	预留	— —	— —	— —	— —	— —	— —	— —	— —
0x1840	PTAPE	PTAPE7	PTAPE6	PTAPE5	PTAPE4	PTAPE3	PTAPE2	PTAPE1	PTAPE0
0x1841	PTASE	PTASE7	PTASE6	PTASE5	PTASE4	PTASE3	PTASE2	PTASE1	PTASE0
0x1842	PTADS	PTADS7	PTADS6	PTADS5	PTADS4	PTADS3	PTADS2	PTADS1	PTADS0
0x1843	预留	—	—	—	—	—	—	—	—
0x1844	PTBPE	PTBPE7	PTBPE6	PTBPE5	PTBPE4	PTBPE3	PTBPE2	PTBPE1	PTBPE0
0x1845	PTBSE	PTBSE7	PTBSE6	PTBSE5	PTBSE4	PTBSE3	PTBSE2	PTBSE1	PTBSE0
0x1846	PTBDS	PTBDS7	PTBDS6	PTBDS5	PTBDS4	PTBDS3	PTBDS2	PTBDS1	PTBDS0
0x1847	预留	—	—	—	—	—	—	—	—
0x1848	PTCPE	0	PTCPE6	PTCPE5	PTCPE4	PTCPE3	PTCPE2	PTCPE1	PTCPE0
0x1849	PTCSE	0	PTCSE6	PTCSE5	PTCSE4	PTCSE3	PTCSE2	PTCSE1	PTCSE0
0x184A	PTCDS	0	PTCDS6	PTCDS5	PTCDS4	PTCDS3	PTCDS2	PTCDS1	PTCDS0
0x184B	预留	—	—	—	—	—	—	—	—
0x184C	PTDPE	PTDPE7	PTDPE6	PTDPE5	PTDPE4	PTDPE3	PTDPE2	PTDPE1	PTDPE0
0x184D	PTDSE	PTDSE7	PTDSE6	PTDSE5	PTDSE4	PTDSE3	PTDSE2	PTDSE1	PTDSE0
0x184E	PTDDS	PTDDS7	PTDDS6	PTDDS5	PTDDS4	PTDDS3	PTDDS2	PTDDS1	PTDDS0
0x184F	预留	—	—	—	—	—	—	—	—
0x1850	PTEPE	PTEPE7	PTEPE6	PTEPE5	PTEPE4	PTEPE3	PTEPE2	PTEPE1	PTEPE0
0x1851	PTESE	PTESE7	PTESE6	PTESE5	PTESE4	PTESE3	PTESE2	PTESE1	PTESE0
0x1852	PTEDS	PTEDS7	PTEDS6	PTEDS5	PTEDS4	PTEDS3	PTEDS2	PTEDS1	PTEDS0
0x1853	预留	—	—	—	—	—	—	—	—
0x1854	PTFPE	PTFPE7	PTFPE6	PTFPE5	PTFPE4	PTFPE3	PTFPE2	PTFPE1	PTFPE0
0x1855	PTFSE	PTFSE7	PTFSE6	PTFSE5	PTFSE4	PTFSE3	PTFSE2	PTFSE1	PTFSE0
0x1856	PTFDS	PTFDS7	PTFDS6	PTFDS5	PTFDS4	PTFDS3	PTFDS2	PTFDS1	PTFDS0
0x1857	预留	—	—	—	—	—	—	—	—
0x1858	PTGPE	0	PTGPE6	PTGPE5	PTGPE4	PTGPE3	PTGPE2	PTGPE1	PTGPE0
0x1859	PTGSE	0	PTGSE6	PTGSE5	PTGSE4	PTGSE3	PTGSE2	PTGSE1	PTGSE0
0x185A	PTGDS	0	PTGDS6	PTGDS5	PTGDS4	PTGDS3	PTGDS2	PTGDS1	PTGDS0
0x185B– 0x185F	预留	— —	— —	— —	— —	— —	— —	— —	— —

1 这个预留位必须总是为 0。

表 4-4 中的非易失性 flash 寄存器，位于 flash 存储器中。这些寄存器中包括一个 8 字节后门密钥，可选用于访问安全的存储器资源。在复位过程中，FLASH 存储器的非易失性寄存器区中的 NVPROT 和 NVOPT 内容会转移到高位页面寄存器中相应的 FPROT 和 FOPT 寄存器，以控制安全和块保护选项。

表 4-4. 非易失性寄存器一览

地址	寄存器名称	Bit 7	6	5	4	3	2	1	Bit 0
$FFB0 – $FFB7	NVBACKKEY	8 字节比较密钥							
$FFB8 – $FFBB	预留	—	—	—	—	—	—	—	—
$FFBC	预留用于存储 250 kHz ICGTRM 值	—	—	—	—	—	—	—	—
$FFBD	NVPROT	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FPDIS
$FFBE	预留用于存储 243 kHz ICGTRM 值	—	—	—	—	—	—	—	—
$FFBF	NVOPT	KEYEN	FNORED	0	0	0	0	SEC01	SEC00

如果密钥使能 （KEYEN）位为 1，那么可用 8 字节对比密钥来临时摆脱存储器安全的限制。这种密钥机制只能通过在安全的存储器中运行的用户代码来使用。（安全密钥不能通过后台调试指令直接进入。）这个安全密钥可通过将 KEYEN 位设为 0 来关闭。如果这个安全密钥关闭，那么摆脱安全限制的唯一方式是完全擦除 Flash （通常通过后台调试接口）并确认 Flash 为空。为了避免在下一次复位后返回到安全模式，应该将安全位 （SEC01:SEC00）设置为非安全状态 （1：0）。

4.3 RAM

MC9S08AC60 系列 包含静态 RAM。 RAM 中 0x0100 以下的位置可以通过更高效的直接寻址模式访问，而这一区域中的任何单个位可以通过位操作指令 （BCLR、 BSET、 BRCLR 和 BRSET）访问。优先将最常访问的编程变量放在 RAM 的这个区域中。

在 MCU 处于低功率等待、停止 2 或停止 3 模式时， RAM 会保留数据。加电启动时， RAM 中的内容不会初始化。如果电源电压没有降到保留 RAM 的最低值以下，则任何复位对 RAM 数据不会产生影响。

为了实现与 M68HC05 MCU 的兼容性，HCS08 将堆栈指针复位到 0x00FF。在系列中，最好将堆栈指针重新初始化到 RAM 顶部，这样，访问频率比较高的 RAM 变量和位可寻址程序变量可以使用直接页面 RAM。将下面这个两指令序列包含在你的复位初始化例程 （其中的 RamLast 等于飞思卡尔半导体等同文件中 RAM 的最高地址）中。

```
        LDHX     #RamLast+1     ;point one past RAM
        TXS                     ;SP<-(H:X-1)
```

当安全性启动时，RAM 被认为是一种安全的存储器资源，不能通过 BDM 或从非安全存储器中执行代码来访问。有关安全特性的更多描述，请参见 4.5 节 安全性。

4.4 FLASH

Flash 存储器主要用作存储程序。在线编程允许在最终组装完应用程序产品后将操作程序加载到 Flash 存储器中。可以通过单线后台调式接口编程整个阵列。由于 Flash 擦除和编程操作都不需要专用电压，通过其它软件控制的通信路径可以实现应用中编程。有关在线编程和在应用中编程的更多讨论，请参阅 《HCS08 系列参考手册》，第 1 卷，飞思卡尔半导体文档订购号 HCS08RMv1。

4.4.1 特性

Flash 存储器的特性包括：

- Flash 大小
  - MC9S08AC60 — 61268 字节 (120 页 ，每页 512 字节)
  - MC9S08AC48 — 49152 字节 (96 页，每页 512 字节)
  - MC9S08AC32 — 32768 字节 (64 页，每页 512 字节)
- 单电源编程和擦除
- 命令介面，实现快速编程和擦除操作
- 典型电压和温度下高达 100,000 编程 / 擦除循环
- 灵活的块保护
- Flash 和 RAM 的安全特性
- 对于低频率读操作自动降低功率

4.4.2 编程和擦除时间

在接受任何编程或擦除命令前，必须写入 Flash 时钟分频寄存器 (FCDIV) ，将 Flash 模块的内部时钟设置为 150 kHz ～ 200 kHz 之间的频率 （f_{FCLK}）(参见 4.6.1 节 Flash 时钟分频寄存器 (FCDIV))。这个寄存器只能写入一次，因此通常是在复位初始化过程中进行写。如果 FSTAT 中的 FACCERR 被设置，即访问错误标记，则不能写入 FCDIV。用户必须确保在写入 FCDIV 寄存器之前没有设置 FACCERR。命令处理器以所得时钟 ($1/f_{FCLK}$) 的一个周期来对编程和擦除脉冲定时。命令处理器用整数个定时脉冲来完成编程或擦除命令。

表 4-5 给出了编程和擦除时间。总线时钟频率和 FCDIV 决定 FCLK 的频率 （f_{FCLK}）。一个 FCLK 周期为 $t_{FCLK} = 1/f_{FCLK}$。显示的时间为多个 FCLK 循环，作为 t_{FCLK} = 5 s 时的一个绝对时间 。显示的编程和擦除时间包括命令状态机的开销及使能和关闭编程和擦除电压的时间。

表 4-5. 编程和擦除时间

参数 r	FCLK 周期	FCLK = 200 kHz 时的时间
字节编程	9	45 μs
字节编程 (突发)	4	20 μs[1]
页面擦除	4000	20 ms
完全擦除	20,000	100 ms

[1] 不包括开始 / 结束开销

4.4.3 编程和擦除命令的执行

FCDIV 寄存器必须初始化，而且在开始执行命令之前，复位和错误标记都要清除。命令执行步骤如下：

1. 将一个数据值写入到 Flash 阵列中的一个地址中。写入的地址和数据信息被锁存在 Flash 接口中。写入操作是任何命令序列要求的第一步。对于擦除和空白检查命令，这些数据的值并不重要。对于页面擦除命令，地址可以是要擦除的 Flash 的 512 字节页面中的任何地址。对于完全擦除和空白检查命令，地址可以是 Flash 存储器中的任何地址。整个 512 字节页面是可被擦除的最小 Flash 单元块。在 60 K 版本中，有两种情况的用户可访问单元块大小低于 512 字节，第一种情况是 RAM 后的第一个页面，第二种情况是高位页面寄存器后的第一个页面。这些页面分别与 RAM 和高位页面寄存器重叠。

注意

成功进行擦除操作后，在 Flash 中编程任何字节的次数不要超过一次。不允许对已经编程的字节进行重新编程，除非先擦除该字节所在的页面，或完全擦除整个 Flash。不先擦除就编程会扰乱 Flash 中存储的数据。

2. 将命令代码写入到 FCMD 中。五个有效的命令分别是空白检查 （ 0x05）、字节编程 （0x20）、突发编程 （0x25）、页面擦除 （0x40）和完全擦除 （ 0x41）。命令代码锁存在命令缓冲器中。
3. 将 FSTAT 中的 FCBEF 位写为 1，清除 FCBEF，并发起 命令 （包括它的地址和数据信息）。

在写入存储器阵列后，写入 1 清除 FCBEF 和发起完整的命令之前，可以通过写 0 到 FCBEF 手动取消部分命令序列。以这种方式取消一个命令，会置 FACCERR 访问错误标记，在启动新的命令前必须清除这个标记。

必须严格遵守监控流程，否则命令不会被接受。这样，可以降低 Flash 存储器内容被意外修改的可能性。FCCF （命令完成标记）指示命令完成的时间。清除 FCBEF ，可以完成命令序列，并发起该命令。图 4-2 为执行所有命令 （突发编程除外）的流程图。在使用任何 Flash 命令前， FCDIV 寄存器必须初始化，在复位以后只能执行一次。

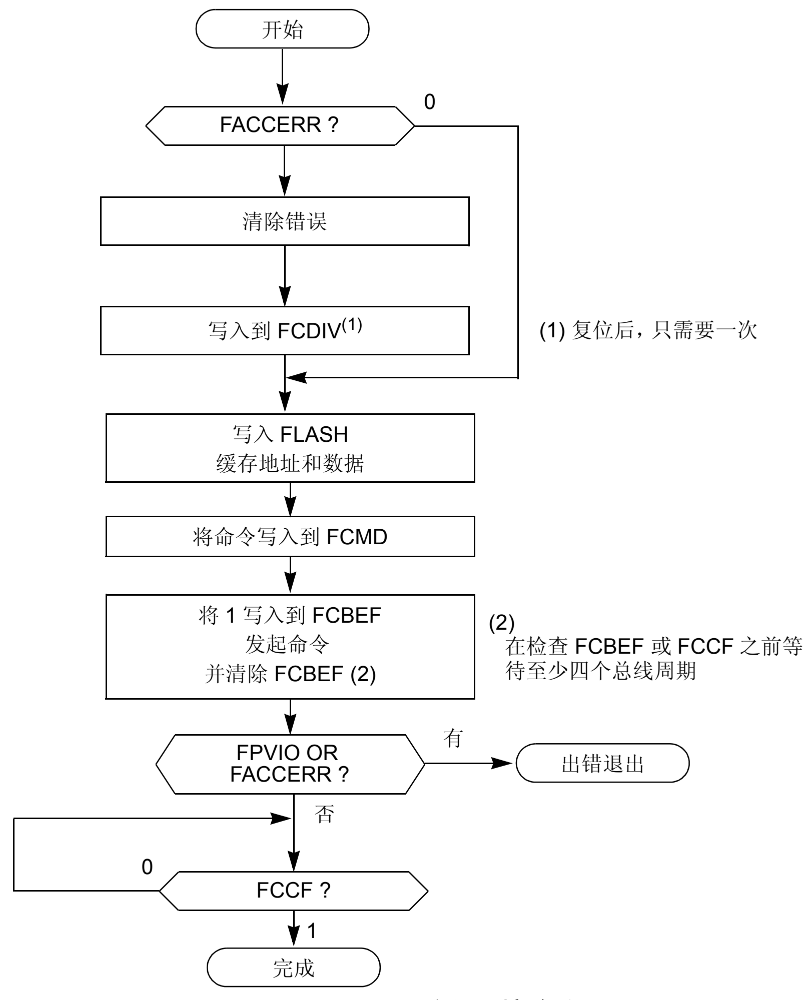


图 4-2. FLASH 编程和擦除流程图

4.4.4 突发编程执行

突发编程命令用来对数据的连续字节进行编程，所用的时间比标准编程命令更短。这是因为编程操作之间不需要关闭 Flash 阵列的高电压。通常情况下，在发出编程或擦除命令后，必须使能一个与 Flash 相关的内部电荷泵，为阵列提供高电压。命令完成后，该电荷泵会关闭。发出突发编程命令后，如果满足以下两个条件，电荷泵会开启而且在突发编程操作完成后将保持开启状态：

1. 下一个突发编程命令在当前编程操作完成前已排队。
2. 下一个顺序地址选择与已编程的当前字节相同的物理行上的一个字节。 Flash 存储器的一行包括 64 个字节。地址 A5 至 A0 选择一行中的一个字节。当地址 A5 至 A0 都为零时，就开始一个新行。

在突发模式下对一系列已经编程的连续字节的第一个字节进行编程，所需要的时间与在标准模式下相同。如果达到上述两个条件，后面的字节将在突发编程时间内编程。如果下一个地址是新的一行的开始，那么该字节的编程时间将是标准时间而不是突发时间。这是因为阵列的高电压必须先关闭然后重新开启。如果在当前命令完成前，队列中没有任何新的突发命令，那么电荷泵将关闭，高电压将从阵列上断开。

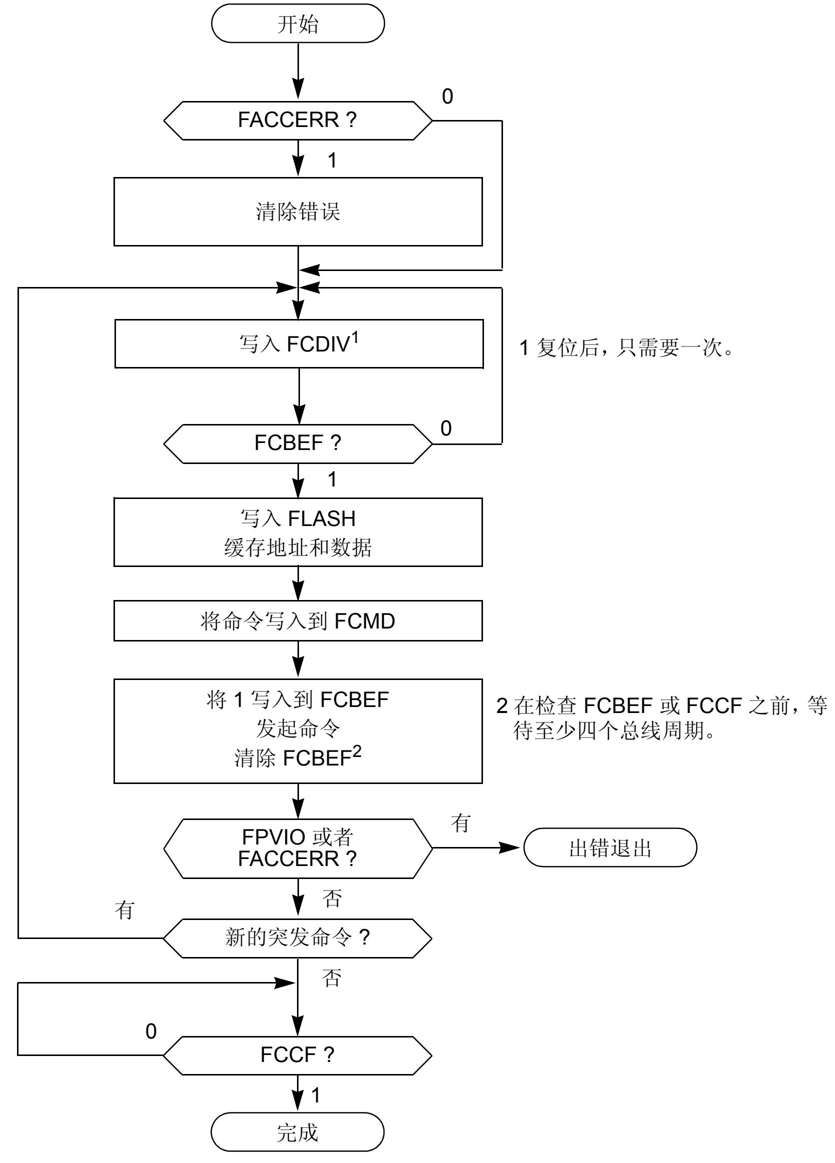


图 4-3. FLASH 突发编程流程图

4.4.5 访问错误

当违反命令执行协议时会出现访问错误。

下列行为将导致 FSTAT 中的访问出错标识位 （FACCERR）被置位。只有当 FACCERR 被清零 （通过写 1 到 FSTAT 的 FACCERR 位）后，才能处理命令

- 在写入 FCDIV 寄存器来设置内部 Flash 时钟频率前，写入 Flash 地址。
- 在没有设置 FCBEF 的情况下写入 Flash 地址 （命令缓冲器清空之前不能开始新的命令）。
- 在发出前一个命令之前第 2 次写入 Flash 地址 （每个命令只能向 Flash 中写入一次）。
- 发出前一个命令之前第 2 次写入 FCMD （每个命令只能向 FCMD 中写入一次）。
- 在写入 Flash 地址后写入 FCMD 以外的任何 Flash 控制寄存器。
- 向 FCMD 写入 5 个允许的代码 （0x05, 0x20, 0x25, 0x40, 0x41, or 0x47）以外的任何命令代码

- 向 FCMD 写入命令之后写除 FSTAT 外的任何 Flash 控制寄存器 （以清除 FCBEF 并发起命令）。
- MCU 在某个编程或擦除命令执行过程中进入停止模式 （命令被终止）。
- MCU 受到安全保护时通过后台调试命令写入字节编程、突发编程、页面擦除命令码 (0x20, 0x25, 或 0x40)。（ 在 MCU 安全时，后台调试控制器只能执行空白检查和完全擦除命令。）
- 将 0 写入到 FCBEF 中，取消部分命令。

4.4.6 Flash 块保护

块保护特性可以防止指定地址范围内的 Flash 存储器发生编程或擦除修改。只要任何 FLASH 块受保护时，全部擦除去活。MC9S08AC60 系列允许 FLASH 末端和 / 或整个 FLASH 存储器的存储器块实施块保护。去活控制位和 3 位控制字段用于设置块的大小。另有一个单独的控制位用于对整个 FLASH 存储器阵列实施块保护。所有这七个控制位都位于 FPROT 寄存器中 （参见 4.6.4 节 FLASH 保护寄存器 （FPROT 和 NVPROT））。

在复位时，NVPROT 位置的内容会加载到高位页面寄存器 （FPROT），这个内容位于 Flash 的非易失性寄存器块中。应用软件不能直接更改 FPROT，因此失控程序不会更改块保护设置。如果包含 NVPROT 的 Flash 的最后 512 字节受到保护，应用程序就无法更改块保护设置 （不管是有意的还是无意的）。若要擦除受保护的 Flash，可通过后台调试命令写入 FPROT 控制位 。

块保护的一个用途是保护 Flash 存储器用于 bootloader 程序的区域。 Bootloader 程序可以擦除 Flash 存储器的其他部分，并重新编程。由于 bootloader 受到保护，即使在擦除和编程操作中 MCU 电源中断，它也不会受到影响。

4.4.7 向量重定向

块保护启动时，复位和中断向量将受到保护。向量重定向使用户可以修改中断向量信息而不需要对 bootloader 和复位向量空间进行解除保护。将位于地址 0xFFBF 的 NVOPT 寄存器中的 FNORED 位设置为 0，可以启动向量重定向。要使重定向发生，必须对地址 $FFBD 上的 NVPROT 寄存器进行编程，使至少部分但不是全部的 Flash 存储器具备块保护。所有中断向量 （存储器位置为 $FFC0–$FFFD）都被重定向，但是复位向量 （$FFFE:FFFF）不重定向。当被保护的超过 32K 时，不能使能向量重定位。

例如，如果 Flash 的 512 个字节受到保护，那么受保护的地址区域为 $FE00 到 $FFFF。中断向量 ($FFC0–$FFFD) 被重定向到位置 $FDC0–$FDFD。现在，如果以 SPI 中断为例，那么，用于向量的值是 $FDE0:FDE1 中的值，而不是 $FFE0:FFE1 中的值。这样，用户就可以利用新的程序代码，包括新的中断向量值对 Flash 的未保护部分进行重新编程，同时使受保护区域，包括缺省向量位置不发生改变。

4.5 安全性

MC9S08AC60 系列包含防止非法访问 Flash 和 RAM 存储器内容的电路。使能了安全功能后， Flash 和 RAM 被看作是安全的资源。直接页面寄存器、高位页面寄存器和后台调试控制器被看作是不安全的资源。安全存储器中执行的程序可以正常访问 MCU 所有存储器的位置和资源。非安全存储器空间执行的程序或通过后台调试接口来访问安全存储器位置的任何尝试都将被阻拦 （写入操作被忽略，而读取操作则全部返回 0）。

安全性的使能和关闭由 FOPT 寄存器中的两个非易失性寄存器位 (SEC01:SEC00]) 的状态确定。在复位过程中，非易失性位置 NVOPT 的内容从 Flash 中拷贝到高位页面寄存器空间内的工作 FOPT 寄存器上。用户可以通过编程 NVOPT 位置来使能安全性。这可以在对 Flash 进行编程的同时完成。如果是 1:0 状态，则安全性关闭；而另外 3 种组合都会使能安全性。请注意，擦除状态时 (1:1) ， MCU 是安全的。在开发过程中，只

要 Flash 被擦除，最好立即将 NVOPT 中的 SEC0 位设置为 0 ，这样 SEC01:SEC00 = 1:0。这将使 MCU 在后续的复位后仍处于非安全状态。

MCU 处于安全状态时不能使能片上调试模块。单独的后台调试控制器仍可用于后台存储器访问命令，但 MCU 不能进入激活后台模式，除非在复位的上升沿使 BKGD 保持低电平。

用户可以选择通过一个 8 字节后门安全密钥来设置允许或不允许安全解锁机制。如果 NVOPT/ FOPT 中的非易失性 KEYEN 位为 0，那么后门密钥关闭，只有通过完全擦除所有 Flash 位置才能解除安全性。如果 KEYEN 为 1，那么安全的用户程序可以通过以下方式来临时关闭安全性：

1. 将 FCNFG 寄存器中的 KEYACC 置于 1。这将使 Flash 模块将写入后门对比密钥位置 (NVBACKKEY 至 NVBACKKEY+7) 的操作解释为与密钥进行比较的值，而不是 Flash 编程或擦除命令的第一步。
2. 将用户输入的密钥值写入到 NVBACKKEY 到 NVBACKKEY+7 位置上。这些写入操作必须按顺序进行，以 NVBACKKEY 的值开始，以 NVBACKKEY+7 的值结束。这些写入操作中不能使用 STHX，因为这些写入不能在相邻的总线周期上完成。用户软件一般通过通信接口（如串行 I/O）从 MCU 系统外部获取密钥代码。
3. 将 FCNFG 寄存器中的 KEYACC 置于 0。如果写入的 8 字节密钥与 Flash 位置上保存的密钥相匹配，那么 SEC01:SEC00 被自动修改为 1:0，同时安全性将关闭，直到下一次复位。

安全密钥只能从 RAM 写入，因此在没有安全的用户程序配合的情况下不能通过后台命令输入。若 KEYACC 位被置位，不能读 FLASH。

后门对比密钥 (NVBACKKEY 至 NVBACKKEY+7) 保存在非易失性寄存器空间内的 Flash 存储器位置中，因此用户可以准确地编程这些位置，就象编程任何其他 Flash 存储器位置一样。非易失性寄存器与复位和中断向量均在同一个 512 字节 Flash 块中，因此对这一空间进行块保护同时也可以保护后门对比密钥。块保护不能通过用户应用程序修改，因此，如果向量空间受到块保护，后门安全密钥机制就不能永久性地修改块保护、安全设置以及后门密钥。

根据以下步骤，可以通过后台调试接口关闭安全性：

1. 通过写入 FPROT 来关闭所有块保护。 FPROT 只能通过后台调试命令写入而不能通过应用软件写入。
2. 如有必要，完全擦除 Flash。
3. 对 Flash 进行空白检查。如果 Flash 完全擦除，那么在下一次复位前安全性一直处于关闭状态。

为了避免在下一次复位后返回到安全模式，对 NVOPT 进行编程使 SEC01:SEC00 = 1:0。

4.6 Flash 寄存器和控制位

Flash 模块在高位页面寄存器空间内有 9 个 8 位寄存器，其中三个位于 Flash 存储器的非易失性寄存器空间，在复位时复制为三个对应的高位页面控制寄存器。 Flash 存储器中还有一个 8 字节对比密钥。对于 所有 Flash 寄存器的绝对地址分配情况，请参见 表 4-3 和 表 4-4。本章节中的寄存器和控制位只提到了名称。通常要用飞思卡尔半导体提供的等同或头文件来将这些名称转换为相应的绝对地址。

4.6.1 Flash 时钟分频寄存器 (FCDIV)

该寄存器的第 7 位是一个只读状态标记。6 至 0 位可以在任何时候读取，但只能写入一次。在开始任何擦除或编程操作之前，写入该寄存器，将非易失性存储器系统的时钟频率设置在可接受的限度内。

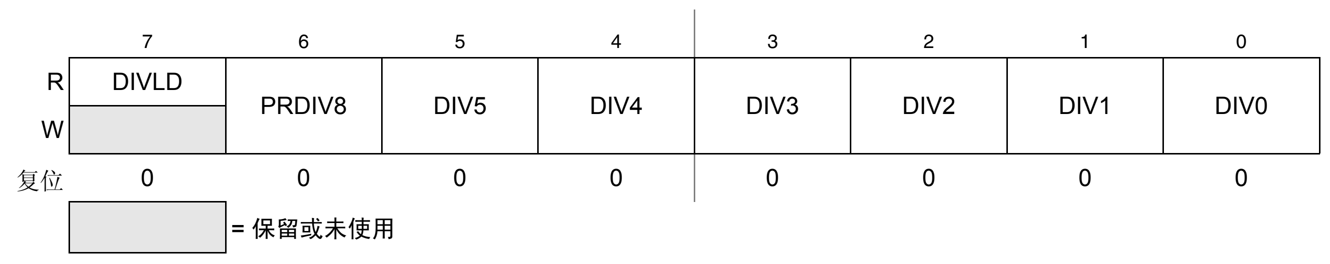


图 4-4. Flash 时钟分频寄存器 (FCDIV)

表 4-6. FCDIV 字段描述

字段	描述
7 DIVLD	**除数加载状态标记** — 设置后，这个只读状态标识 FCDIV 寄存器自复位后已经被写入。复位会清除该位，第一次写入该寄存器的操作将导致该位被设置，不管写入什么数据。 0 FCDIV 自复位后没有被写入； Flash 的擦除和编程操作关闭。 1 FCDIV 自复位后已写入； Flash 的擦除和编程操作已使能。
6 PRDIV8	**预分频 （分频）Flash 时钟除以 8** 0 输入到 Flash 时钟分频器的时钟为总线频率时钟。 1 输入到 Flash 时钟分频器的时钟为总线频率时钟除以 8。
5 DIV[5:0]	**Flash 时钟分频器除数** — 时钟分频器用 6 位 DIV5:DIV0 字段中的值加 1 除总线频率时钟 （如果 PRDIV8 = 1，则用总线频率时钟除以 8）。得出的内部 Flash 时钟的频率必须在 200 kHz 到 150 kHz 的范围内，这样才能使 Flash 正常运行。编程 / 擦除定时脉冲为这个内部 Flash 时钟的一个周期，这相当于 5 ms 到 6.7 ms。自动编程逻辑使用整数个这些脉冲来完成擦除或编程操作。请参见 公式 4-1 和 公式 4-2。表 4-7 显示了所选总线频率的 PRDIV8 和 DIV5:DIV0 的相应值。

$$\text{如果 PRDIV8 = 0 — } f_{FCLK} = f_{Bus} \div ([DIV5:DIV0] + 1) \qquad \textit{公式 4-1}$$

$$\text{如果 PRDIV8 = 1 — } f_{FCLK} = f_{Bus} \div (8 \times ([DIV5:DIV0] + 1)) \qquad \textit{公式 4-2}$$

表 4-7. FLASH 时钟分频器设置

f_{Bus}	PRDIV8 (二进制)	DIV5:DIV0 (十进制)	f_{FCLK}	编程 / 擦除定时脉冲 (最小 5 μs, 最大 6.7 μs)
20 MHz	1	12	192.3 kHz	5.2 μs
10 MHz	0	49	200 kHz	5 μs
8 MHz	0	39	200 kHz	5 μs
4 MHz	0	19	200 kHz	5 μs
2 MHz	0	9	200 kHz	5 μs
1 MHz	0	4	200 kHz	5 μs
200 kHz	0	0	200 kHz	5 μs
150 kHz	0	0	150 kHz	6.7 μs

4.6.2 Flash 选项寄存器 （FOPT 和 NVOPT）

在复位过程中，非易失性位置 NVOPT 的内容从 Flash 拷贝到 FOPT 中。第 5 位至第 2 位不使用，读为 0。这个寄存器可以在任何时候读取，但写入无任何意义或影响。若想修改这个寄存器中的值，可对 Flash 中的 NVOPT 位置进 Flash 擦除和重新编程，然后发出新的 MCU 复位命令。

	7	6	5	4	3	2	1	0
R	KEYEN	FNORED	0	0	0	0	SEC01	SEC00
W								
复位	在复位期间，这个寄存器从非易失性置 NVOPT 中加载 。 .							

= 保留或未使用

图 4-5. FLASH 选项寄存器 （FOPT）

表 4-8. FOPT 字段描述

字段	描述
7 KEYEN	**后门密钥机制使能** — 该位设置为 0 时，不能使用后门密钥机制来关闭安全性。后门密钥机制只能从用户 （安全的）固件上访问。不能用 BDM 命令来写入可能会打开后门密钥的密钥对比值。有关后门密钥机制的更多信息，请参见 4.5 节 安全性。 0 不允许后门密钥访问。 1 如果用户固件写入一个与非易失性后门密钥 （从 NVBACKKEY 到 NVBACKKEY+7 按顺序）相匹配的 8 字节的值，安全性在下一次 MCU 复位前会暂时关闭。
6 FNORED	**向量重定向关闭** — 该位为 1 时向量重定向关闭。 0 向量重定向使能。 1 向量重定向关闭
1:0 SEC0[1:0]	**安全状态代码** — 这个 2 位字段决定 MCU 的安全状态，如下所示。MCU 处于安全状态时，RAM 和 Flash 中的内容不能通过指令从不安全的源 （包括后台调试接口）访问。有关安全性的详细信息，请参见 4.5 节 安全性 . 00 安全 01 安全 10 不安全 11 安全 在成功输入后门密钥或成功进行 flash 空白校验后， SEC0[1:0] 改为 10。

4.6.3 FLASH 配置寄存器 （FCNFG）

第 5 位可以随时读写。其余总是可读，但不能写。

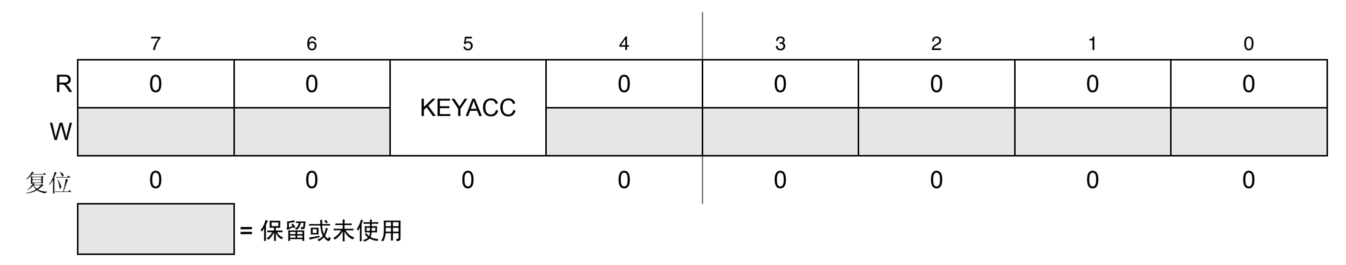


图 4-6. FLASH 配置寄存器 （FCNFG）

表 4-9. FCNFG 字段描述

字段	描述
5 KEYACC	**使能访问密钥的写入** — 该位启用后门对比密钥的写入。有关后门密钥机制的更多信息，请参见 4.5 节 安全性。 0 写入 0xFFB0-0xFFB7 被解释为 Flash 编程或擦除命令的开始。 1 写入 NVBACKKEY (0xFFB0-0xFFB7) 被解释为对比密钥写入。 读 flash 后返回无效数据。

4.6.4 FLASH 保护寄存器 （FPROT 和 NVPROT）

在复位过程中，非易失性位置 NVOPT 的内容从 flash 拷贝到 FPROT 中。这个寄存器可以在任何时候读取，但写入无任何意义或影响。后台调试命令可以写入 FPROT。

	7	6	5	4	3	2	1	0
R	FPS7	FPS6	FPS5	FPS4	FPS3	FPS2	FPS1	FPDIS
W	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)
复位	在复位期间，这个寄存器从非易失性位置 NVOPT 中加载							

1 可以使用后台命令更改 FPROT 中这些位的内容。

图 4-7. Flash 保护寄存器 （FPROT）

表 4-10. FPROT 寄存器字段描述

字段	描述
7:1 FPS[7:1]	**Flash 保护选择位** — FPDIS = 0 时，这个 7 位字段决定位于 Flash 高端地址末尾的未保护 Flash 位置的结束地址。受保护的 Flash 位置不能被擦除或编程。
0 FPDIS	**Flash 保护关闭** 0 FPS[7:1] 指定的 Flash 块具备块保护 （不允许编程和擦除）。 1 Flash 块无保护。

4.6.5 Flash 状态寄存器 （FSTAT）

3、 1 和 0 位总是读为 0，写操作没有意义或影响。剩下的五位是状态位，随时可读。写入这些位有特殊意义，将在位描述中讨论。

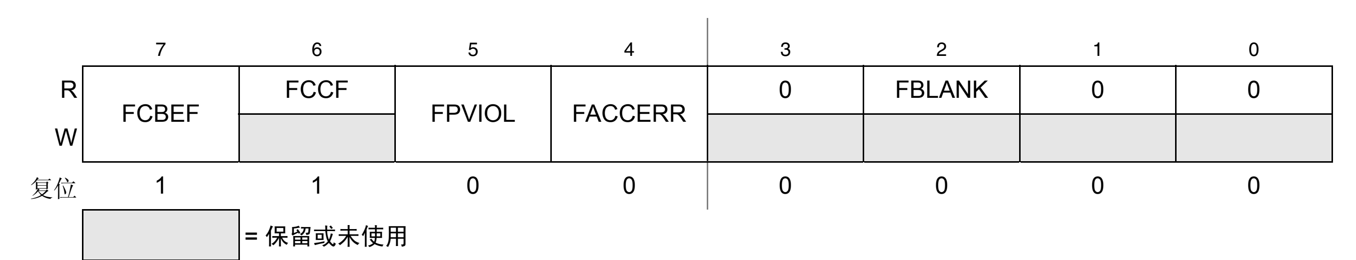


图 4-8. Flash 状态寄存器 （FSTAT）

表 4-11. FSTAT 字段描述

字段	描述
7 FCBEF	**Flash 命令缓冲器空标记** — FCBEF 位用于发出命令。它还指示命令缓冲器是空的，这样可以在执行突发编程时执行新的命令序列。写入 1 或当突发编程命令转移到阵列进行编程时，可以清除 FCBEF 位。只有突发编程命令可以被缓存。 0 命令缓冲器满 （不能容纳额外的命令）。 1 命令缓冲器可写入新的突发编程命令。
6 FCCF	**Flash 命令完成标记** — 命令缓冲器为空而且无命令在处理时自动置 FCCF 。开始执行新命令时自动清除 FCCF （将 1 写到 FCBEF 中以注册一个命令）。写入 FCCF 没有任何意义或影响。 0 命令正在执行过程中。 1 所有命令都已完成。
5 FPVIOL	**保护破坏标记** — 当 FCBEF 被清除以注册一个试图擦除或编程受保护的块中的位置时， FPVIOL 自动设置 （错误的命令会被忽略）。将 FPVIOL 置 1 可清除 FPVIOL 。 0 无保护破坏。 1 有擦除或编程受保护位置的企图。
4 FACCERR	**访问错误标记** — FACCERR 在以下情况下自动设置：没有严格遵守正确的命令序列 （错误的命令将被忽略）， FCDIV 寄存器初始化之前企图进行编程或擦除操作，或在命令正在执行时 MCU 进入停止模式。欲了解哪些行为被认为是访问错误的详细信息，请参见 4.4.5 节 访问错误。通过 FACCERR 置 1 清除 FACCERR。将 0 写入 FACCERR 没有意义或影响。 0 无访问错误。 1 发生了访问错误。
2 FBLANK	**FLASH 被确认为全空（被擦除）标记** — FBLANK 在空白检查命令完成后自动设置为 1（如果整个 Flash 阵列被确认已擦除）。FBLANK 通过清除 FCBEF 以写入新的有效命令来清除。写入 FBLANK 没有任何意义或影响。 0 在空白检查命令执行完成而且 FCCF = 1 的情况下，FBLANK = 0 指示 Flash 阵列未被完全擦除。 1 在空白检查命令执行完成而且 FCCF = 1 的情况下，FBLANK = 1 指示 Flash 或 EEPROM 阵列已完全擦除（全 $FF）。

4.6.6 FLASH 命令寄存器 （FCMD）

如表 4-13 所示，正常用户模式下只能识别 5 种命令代码。有关 flash 编程和擦除操作的详细讨论，请参见 4.4.3 节 编程和擦除命令的执行。

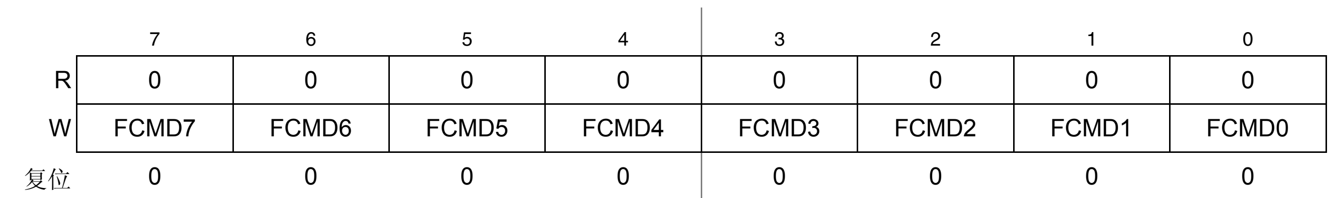


图 4-9. FLASH 命令寄存器 （FCMD）

表 4-12. FCMD 字段描述

字段	描述
7:0 FCMD[7:0]	FCMD[7:0] 的描述参见表 4-13。

表 4-13. Flash 命令

命令	FCMD	等同文件标签
空白检查	$05	mBlank
字节编程	$20	mByteProg
字节编程 — 突发模式	$25	mBurstProg
页面擦除 （512 字节 / 页）	$40	mPageErase
完全擦除 （所有 flash）	$41	mMassErase

所有其它命令代码都为非法代码，生成访问错误。

进行完全擦除操作后没有必要再执行空白检查命令。只有作为安全解锁机制的一部分时才需要空白检查。

第 5 章
复位、中断和系统配置

5.1 介绍

本章讨论了 系列中的基本复位和中断机制，以及各种复位和中断源。外围模块的某些中断源在本参考手册的其他章节中有更详细的讨论。本节收集了所有复位和中断源的基本信息，以便参考。而有些复位和中断源，包括计算机正常操作 （COP）的看门狗和实时中断 （RTI），它们本身不是片上外围系统，但是系统控制逻辑的一部分。

5.2 特性

复位和中断特性包括：

- 多源复位，实现灵活的系统配置和可靠操作
  - 加电检测 (POR)
  - 低压检测 (LVD)，使能
  - 外部 RESET 引脚
  - COP 看门狗使能，及两个超时选择
  - 非法操作代码
  - 来自后台调试主机的串行命令
- 复位状态寄存器 (SRS) ，指示最新复位的源
- 每个模块的单独中断向量 （减少轮询开销） (参见 表 5-11)

5.3 MCU 复位

复位 MCU 是一个从已知初始状况启动处理的方法。复位期间，大部分控制和状态寄存器强制使用初始值，并从复位向量 (0xFFFE:0xFFFF) 加载程序计数器。片上外围模块关闭，I/O 引脚初始配置为上拉器件被关闭的通用高阻抗输入。条件代码寄存器 (CCR) 中的 I 位被设置，用来阻拦可屏蔽中断，以便用户程序对堆栈指针 (SP) 和系统控制设置进行初始化。 SP 在复位时强制设为 0x00FF。

MC9S08AC60 系列有多个复位源：

- 加电复位 （POR）
- 低电压检测 （LVD）
- 计算机正常操作 (COP) 定时器
- 非法操作代码检测
- 后台调试强制复位
- 外部复位引脚 (RESET)
- 时钟发生器时钟丢失和时钟丢失复位

上述复位源（后台调试强制复位除外）在系统复位状态寄存器 (SRS) 中都有一个相关位。只要 MCU 进入复位，内部时钟发生器 (ICG) 模块就切换到自时钟模式，并选择频率为 f_{Self_reset}。内部总线频率为 ICG 频率的一半时，复位引脚的电平降低 34 个总线周期。完成 34 个总线周期后，该引脚被释放，并由内部上拉电阻上拉，直到它在外部保持低电平。该引脚释放，再经过 38 个总线周期后，对它进行采样，确定 MCU 复位是否由复位引脚造成。

5.4 计算机正常操作 (COP) 看门狗

当应用系统软件不能正常执行时，COP 看门狗将强制进行系统复位。为了防止从 COP 定时器（当 COP 定时器使能时）发起系统复位，应用软件必须定期复位 COP 计数器。如果应用程序在超时前未能复位 COP 计数器，这时会生成一个系统复位，强迫系统回到已知起点。

任何复位后， SOPT 中的 COPE 位都会置位来激活 COP 看门狗（更多信息请参见 5.9.4 节 系统选项寄存器 (SOPT)。如果应用中没有使用 COP 看门狗，可以通过清除 SOPT 的 COPT 位进行禁止。在 SRS 地址中写入任何值，可复位 COP 计数器。写入不会影响只读 SRS 中的数据。写入这个地址的操作将被解码，然后发送复位信号给 COP 计数器。

SOPT2 中的 COPCLKS 位 （更多信息请参见 5.9.10 节 系统选项寄存器 2 (SOPT2)）设置供 COP 定时器使用的时钟源。时钟源可以是总线时钟或 1 kHz 内部时钟源。对任一个时钟源来说，都有由 SOPT 中的 COPT 控制的相关长和短超时。表 5-1 列举了 COPCLKS 和 COPT 位的控制功能。COP 看门狗默认总线时钟源，长超时 （218 周期）。

表 5-1. COP 配置选项

控制位		时钟源	COP 溢出计数
COPCLKS	COPT		
0	0	~1 kHz	2^5 周期 (32 ms)[1]
0	1	~1 kHz	2^8 周期 (256 ms)[1]
1	0	Bus	2^{13} 周期
1	1	Bus	2^{18} 周期

[1] 栏中显示的值基于 t_{RTI} = 1 ms。参见附录 A.10.1 节 控制时序 中的 t_{RTI} 值，了解这个值允许的偏差。

即使应用程序使用复位默认的 COPT、COPCLKS 和 COPW 位的设置，用户也必须在复位初始化过程中写入只可写入一次的 SOPT1 和 SOPT2 寄存器上，以便锁存该设置。如果应用程序丢失，使用这种方式可以防止意外修改。初始写入 SOPT 和 SOPT2 会复位 COP 计数器。

写入服务 （清除）COP 计数器的 SRS ，这个操作不能放在中断服务程序 （ISR）中，因为即使是在主应用程序不能正常执行时，仍然可以定期执行 ISR。

MCU 处于后台调试模式时，COP 计数器不会计数。

当选择总线时钟源时，系统如果处于停止模式， COP 计数器不会计数。当 MCU 退出停止模式时， COP 计数器立即恢复计数。

如果选择 1 kHz 时钟源，那么一旦进入停止模式，COP 计数器就会被重新初始化为 0，并在 MCU 退出停止模式时从 0 开始计数。

5.5 中断

中断可保存当前 CPU 状态和寄存器，执行中断服务例程 (ISR)，然后恢复 CPU 状态，使处理恢复到中断前的位置。与软件中断 （SWI）不同 （SWI 由程序指令触发），中断是由诸如 IRQ 引脚边沿或定时器溢出等硬件事件触发。调试模块在特定情况下也会产生 SWI。

如果一个使能的中断源内发生事件，将会设置相关的只读状态标记。但 CPU 不会响应，除非 / 直到本地中断使能设置为逻辑 1。CCR 中的 I 位为 0 ，允许中断。CCR 中的全局中断屏蔽（I 位）在复位后首次设置，会屏蔽所有可屏蔽的中断源。在清除 I 位之前，用户程序初始化堆栈指针，执行其他系统设置，允许 CPU 响应中断。

当 CPU 接收到符合条件的中断请求时，它会在响应中断前先完成当前指令。中断顺序与 SWI 指令的逐个周期顺序相同，这个顺序是：

- 将 CPU 寄存器保存在堆栈上
- 在 CCR 中设置 I 位，屏蔽进一步的中断
- 为当前挂起的最高优先级中断获取中断向量
- 填充指令队列，程序信息的前 3 个字节从获取中断向量位置的地址开始

当 CPU 响应中断时， I 位会自动设置，以避免出现又一个中断来中断 ISR 自身 （这称为中断嵌套）。通常，当 CCR 从进入 ISR 时堆栈的值恢复时，I 位就恢复为 0。在极个别情况中，I 位可以在 ISR 内部清除（在清除生成中断的状态标记后），所以无需等待第一个业务例程完成，就可以执行另一个中断。该操作可能导致一些难以调试的细微程序错误，只建议经验丰富的程序员使用。

中断服务例程以中断恢复 （RTI）指令作为结束。 RTI 指令从堆栈中读取先前保存的信息，将 CCR、A、X 和 PC 寄存器恢复为中断前的值。

注意

为了实现与 M68HC08 器件的兼容，H 寄存器不能自动保存和恢复。建议在中断服务例程 （ISR）开始时就将 H 推到堆栈上，并在使用 RTI 从 ISR 中恢复之前立即恢复它。

I 位被清除时如果有两个以上挂起的中断，则首先处理优先级最高的源 （参见 表 5-2）。

5.5.1 中断堆栈帧

图 5-1 显示了堆栈帧的内容和结构。在中断前，堆栈指针 （SP）指向堆栈的下一个可用字节。CPU 寄存器的当前值保存在堆栈中，以程序计数器 （PCL）的低阶字节开始，以 CCR 结束。堆栈操作后，SP 指向堆栈的下一个可用位置，该堆栈是比保存 CCR 的地址小 1 的地址。堆栈的 PC 值是主程序 （如果不发生中断，则执行主程序）的指令地址。

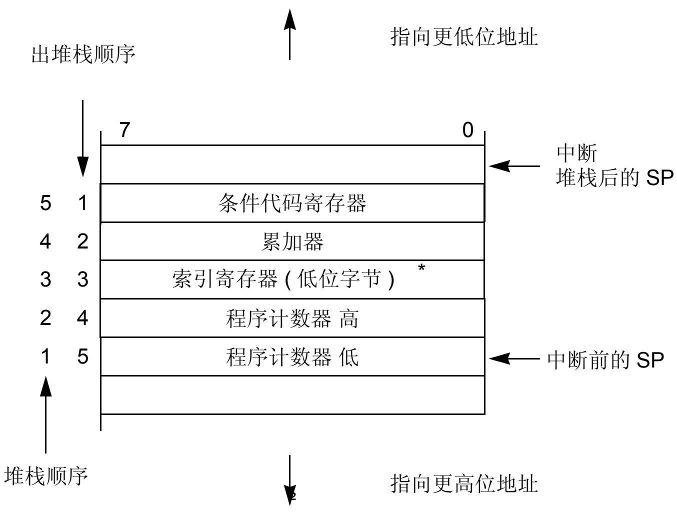


图 5-1. 中断堆栈帧

执行 RTI 指令时，这些值以相反顺序从堆栈中恢复。其中一个步骤是，CPU 读取程序信息的 3 个字节（始于从堆栈中恢复的 PC 地址），填充指令管道。

从 ISR 返回前，必须应答 （清除）导致中断的状态标记。通常，该标记应该在 ISR 开始时被清除，这样如果同一个源生成另外一个中断，就可以注册这个中断，在当前 ISR 完成后再处理这个中断。

5.5.2 外部中断请求 （IRQ）引脚

外部中断由 IRQ 状态和控制寄存器 （IRQSC）管理。当 IRQ 功能开启后，同步逻辑监控引脚是否发生边沿触发或边沿及电平触发事件。当 MCU 处于停止模式且系统时钟关闭时，将使用独立的异步路径，这样 IRQ （如果使能）就可以唤醒 MCU。

5.5.2.1 引脚配置选项

IRQSC 寄存器中的 IRQ 引脚使能 （IRQPE）控制位必须为 1，这样 IRQ 引脚才能作为中断请求 （IRQ）输入。作为 IRQ 输入，用户可以选择被检测的边沿或电平 （IRQEDG）的极性，引脚是否检测到仅边沿触发或是检测到边沿和电平触发 （IRQMOD），或者是否一个事件导致了中断还是只设置了通过软件可以轮询的 IRQF 标记。

IRQ 引脚使能后，默认使用内部上拉 / 下拉器件 （IRQPDD = 0），器件究竟是上拉还是下拉取决于所选的极性。如果用户希望使用外部上拉或下拉，可以在 IRQPDD 中写入 1，以关闭内部器件。

当把 IRQ 引脚配置为 IRQ 输入时，可以使用 BIH 和 BIL 指令来检测 IRQ 引脚上的电平。

注意

- 上拉 IRQ 引脚上测量到的电压可能只有 V_{DD} – 0.7 V。与该引脚相连的内部门一直拉到 V_{DD}。所有使能了上拉电阻的其它引脚都将有未加载的 V_{DD} 测量。
- 当使能 IRQ 引脚供使用时， IRQF 将被设置，并且必须在使能中断前清除。如果在 5V 系统中，将这个引脚配置为下降边沿和电平敏感，有必要在清除这个标记和使能中断之间至少等待 6 个周期。

5.5.2.2 边沿和电平灵敏度

IRQMOD 控制位重新配置检测逻辑，这样它就能检测边沿事件和引脚电平。在边沿检测模式中，当检测到边沿时 （IRQ 引脚从无效改为有效电平时），IRQF 状态标记就被设置，但只要 IRQ 引脚处于有效电平，就会连续设置该标记 （并且不能清除）。

5.5.3 中断向量、源和本地掩码

表 5-2 列出了所有中断源。较高优先级的源位于表格下方。中断服务例程地址的高阶字节位于向量地址栏的第一个地址，中断服务例程地址的低阶字节位于下一个高阶地址。

当出现中断时，相关标记位被设置。如果相关的本地中断使能位是 1，中断请求会发送到 CPU。在 CPU 中，如果全局中断屏蔽（CCR 中的 I 位）为 0，CPU 将完成当前指令；堆栈 PCL、PCH、X、A 和 CCR CPU 寄存器；并设置 I 位；然后为挂起的最高优先级中断获取中断向量。然后继续处理中断服务例程。

表 5-2. 向量摘要

向量优先级	向量编号	地址（高 / 低）	向量名称	模块	源	使能	描述
低	29 – 31	0xFFC0/FFC1 – 0xFFC4/0xFFC5	未使用的向量空间（提供给用户程序）				
	28	0xFFC6/FFC7	Vtpm3ovf	TPM3	TOF	TOIE	TPM3 溢出
	27	0xFFC8/FFC9	Vtpm3ch1	TPM3	CH1F	CH1IE	TPM3 通路 1
	26	0xFFCA/FFCB	Vtpm3ch0	TPM3	CH0F	CH0IF	TPM3 通路 0
	25	0xFFCC/FFCD	Vrti	System control	RTIF	RTIE	实时中断
	24	0xFFCE/FFCF	Viic1	IIC1	IICIF	IICIE	IIC1
	23	0xFFD0/FFD1	Vadc1	ADC1	COCO	AIEN	ADC1
	22	0xFFD2/FFD3	Vkeyboard 1	KBI1	KBF	KBIE	KBI1 引脚
	21	0xFFD4/FFD5	Vsci2tx	SCI2	TDRE, TC	TIE, TCIE	SCI2 发送
	20	0xFFD6/FFD7	Vsci2rx	SCI2	IDLE, RDRF, LDBKDIF, RXEDGIF	ILIE, RIE, LBKDIE, RXEDGIE	SCI2 接收
	19	0xFFD8/FFD9	Vsci2err	SCI2	OR, NF, FE, PF	ORIE, NFIE, FEIE, PFIE	SCI2 错误
	18	0xFFDA/FFDB	Vsci1tx	SCI1	TDRE TC	TIE TCIE	SCI1 发送
	17	0xFFDC/FFDD	Vsci1rx	SCI1	IDLE, RDRF, LDBKDIF, RXEDGIF	ILIE, RIE, LBKDIE, RXEDGIE	SCI1 接收
	16	0xFFDE/FFDF	Vsci1err	SCI1	OR, NF, FE, PF	ORIE, NFIE, FEIE, PFIE	SCI1 错误
	15	0xFFE0/FFE1	Vspi1	SPI1	SPIF, MODF, SPTEF	SPIE, SPIE, SPTIE	SPI1
	14	0xFFE2/FFE3	Vtpm2ovf	TPM2	TOF	TOIE	TPM2 溢出
	13	0xFFE4/FFE5	Vtpm2ch1	TPM2	CH1F	CH1IE	TPM2 通路 1
	12	0xFFE6/FFE7	Vtpm2ch0	TPM2	CH0F	CH0IE	TPM2 通路 0
	11	0xFFE8/FFE9	Vtpm1ovf	TPM1	TOF	TOIE	TPM1 溢出
	10	0xFFEA/FFEB	Vtpm1ch5	TPM1	CH5F	CH5IE	TPM1 通路 5
	9	0xFFEC/FFED	Vtpm1ch4	TPM1	CH4F	CH4IE	TPM1 通路 4
	8	0xFFEE/FFEF	Vtpm1ch3	TPM1	CH3F	CH3IE	TPM1 通路 3
	7	0xFFF0/FFF1	Vtpm1ch2	TPM1	CH2F	CH2IE	TPM1 通路 2
	6	0xFFF2/FFF3	Vtpm1ch1	TPM1	CH1F	CH1IE	TPM1 通路 1
	5	0xFFF4/FFF5	Vtpm1ch0	TPM1	CH0F	CH0IE	TPM1 通路 0
	4	0xFFF6/FFF7	Vicg	ICG	ICGIF (LOLS/LOCS)	LOLRE/LOCRE	ICG
	3	0xFFF8/FFF9	Vlvd	System control	LVDF	LVDIE	低压检测
	2	0xFFFA/FFFB	Virq	IRQ	IRQF	IRQIE	IRQ 引脚
	1	0xFFFC/FFFD	Vswi	Core	SWI Instruction	—	软件中断
Higher	0	0xFFFE/FFFF	V 复位	System control	COP LVD $\overline{\text{RESET}}$ 引脚 非法操作码	COPE LVDRE — —	看门狗定时器 低压检测 外部引脚 非法操作码

5.6 低电压检测 (LVD) 系统

MC9S08AC60 系列 具有防止低电压的系统，在电源电压不稳时能够保护存储器内容和控制 MCU 系统状态。该系统由加电复位 （POR）电路和 LVD 电路组成，其中 LVD 电路带用户可选择的跳变电压，高压 (V_{LVDH}) 或低压 (V_{LVDL})。当 SPMSC1 中的 LVDE 为高 并且 SPMSC2 中的 LVDV 选择了跳变电压，LVD 电路使能。进入停止模式时， LVD 关闭，除非设置了 LVDSE 位。如果同时设置了 LVDSE 和 LVDE，那么 MCU 不能进入停止 2 ， 而且 使能 LVD 的停止 3 模式将更耗电。

5.6.1 加电复位操作

当首次接通 MCU 的电源时，或当电源电压低于 V_{POR} 时，POR 电路会发起复位。随着电源电压升高，LVD 电路让 MCU 保持复位状态，直到电源电压高于 V_{LVDL} 。 POR 后，SRS 中 POR 位和 LVD 位同时被设置。

5.6.2 LVD 复位操作

配置 LVD （设置 LVDRE 为 1），使其在检测到低电压情况时发起复位。在发生 LVD 复位后，LVD 系统会让 MCU 保持复位状态，直到电源电压高于 LVDV 确定的阈值。LVD 复位或 POR 后都会在 SRS 寄存器中设置 LVD 位。

5.6.3 LVD 中断操作

当检测到低压情况，且 LVD 电路配置给中断操作 (LVDE 设置，LVDIE 设置， LVDRE 清除)，则 LVDF 将设置，发生 LVD 中断。

5.6.4 低压告警 (LVW)

LVD 系统有一个低压提醒标记，提示电源电压正接近但仍高于 LVD 电压。 LVW 没有相关的中断。 LVW 有两个用户可选择的跳变电压，一个是 高 (V_{LVWH})，一个为低 (V_{LVWL})。跳变电压通过 SPMSC2 中的 LVWV 设置。建议不要将 LVW 跳变电压设置为与 LVD 跳变电压相等。 LVW 的典型作用是选择 V_{LVWH} 和 V_{LVDL}。

5.7 实时中断 (RTI)

实时中断功能可用来生成定期中断。RTI 可以接受两个时钟源， 1 kHz 内部时钟或外部时钟 （如果存在）。1 kHz 内部时钟源完全独立于任何总线时钟源，只用于 RTI 模块，在一些 MCU 上还可用于 COP 看门狗 。要使用外部时钟源，则必须提供且激活外部时钟源。 SRTISC 中的 RTICLKS 位用于选择 RTI 时钟源。

当 MCU 处于运行、等待或停止 3 模式时，可以使用其中任何一个 RTI 时钟源 。当在停止 3 模式使用外部振荡器时，它必须在停止模式中使能 (OSCSTEN = 1) ，并配置为低带宽操作 (RANGE = 0)。只可以选择内部 1 kHz 时钟源，将 MCU 从停止 2 模式唤醒。

SRTISC 寄存器包含只读状态标记、只写应答位和一个 3 位控制值 (RTIS2:RTIS1:RTIS0)，用于关闭实时中断的时钟源或从七个唤醒周期中选择一个。RTI 有一个内部中断使能位， RTIE，来屏蔽实时中断。可以将 RTIS 的每个位写为零，以此来关闭 RTI，不再产生中断。有关该寄存器的更多信息，参见 5.9.7 节 系统实时中断状态和控制寄存器 (SRTISC)。

5.8 MCLK 输出

PTC2 引脚复用于 MCLK 时钟输出。设置引脚的使能位 MPE，会造成 PTC2 引脚输出分频后的内部 MCU 总线时钟。分频比率由 MCSEL 位决定。若设置了 MPE，无论 PTC2 引脚的端口数据方向控制位的状态如何，该引脚都被强制作为输出引脚工作。如果 MCSEL 位均为 0，则该引脚驱向低电平位置。引脚的斜率和驱动强度分别由 PTCSE2 和 PTCDS2 控制。如果斜率控制功能被打开，最大时钟输出频率将会受到限制。如需了解斜率使能时的引脚上升和下降时间，请参见电气技术规范。

5.9 复位、中断及系统控制寄存器和控制位

直接页面寄存器空间里的一个 8 位寄存器和高位页面寄存器空间的八个 8 位寄存器都与复位和中断系统相关。

欲了解各寄存器的绝对地址分配，请参见本文档第 4 章 存储器中的直接页面寄存器一览。本小节只提及寄存器和控制位的名称。可以参阅飞思卡尔提供的等同或头文件将这些名称转换为相应的绝对地址。

SOPT 和 SPMSC2 寄存器中的某些控制位与操作模式有关。本小节仅对这些位进行简要描述，详细的描述参见 第 3 章 操作模式。

5.9.1 中断引脚请求状态和控制寄存器 (IRQSC)

这个直接页面寄存器包括状态和控制位，用来配置 IRQ 功能、报告状态和应答 IRQ 事件。

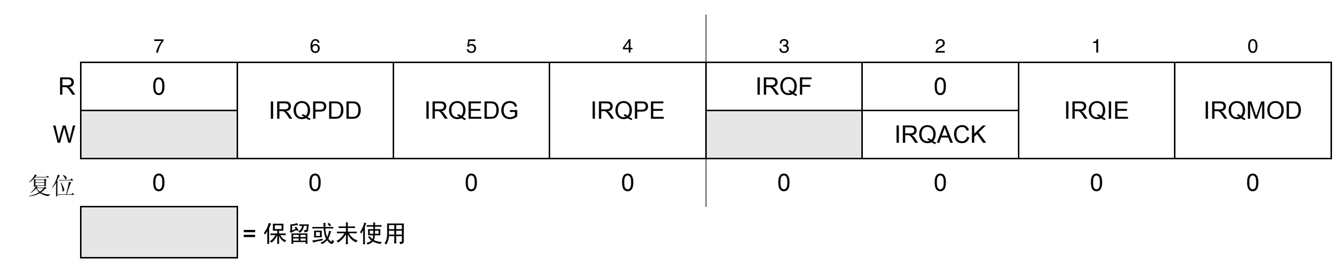


图 5-2. 中断请求状态和控制寄存器 (IRQSC)

表 5-3. IRQSC 寄存器字段描述

字段	描述
6 IRQPDD	**中断请求 (IRQ) 上拉器件关闭** — IRQ 引脚使能时 (IRQPE = 1)，这个读 / 写控制位用来关闭内部上拉器件，允许使用外部器件。 0 如果 IRQPE = 1，IRQ 上拉器件使能。 1 如果 IRQPE = 1，IRQ 上拉器件关闭。
5 IRQEDG	**中断请求 (IRQ) 边沿选择** — 这个读 / 写控制位用来选择用于 IRQF 设置的 IRQ 引脚上的边沿或电平极性。IRQMOD 控制位决定 IRQ 引脚对边沿和电平都敏感还是只对边沿敏感。当 IRQ 引脚作为 IRQ 输入使能且用来检测上升边沿时，这个可选上拉电阻重新配置为可选下拉电阻。 0 IRQ 是下降边沿，或者下降边沿 / 对低电平敏感。 1 IRQ 是上升边沿，或者上升边沿 / 对高电平敏感。
4 IRQPE	**IRQ 引脚使能** — 这个读写控制位启用 IRQ 引脚。当设置了该位时，IRQ 引脚可以用作中断请求。而且，当设置这个位时，可以根据 IRQMOD 位的状态，使能内部上拉或内部下拉电阻。 0 IRQ 引脚功能关闭。 1 IRQ 引脚功能使能。

表 5-3. IRQSC 寄存器字段描述 (续)

字段	描述
3 IRQF	**IRQ 标记** — 当发生了中断请求事件，该只读状态位对此进行标识。 0 无 IRQ 请求。 1 检测到 IRQ 事件。
2 IRQACK	**IRQ 应答** — 这个只写位用来应答中断请求事件 （写入 1 清除 IRQF）。写入 0 则没有任何意义或影响。读总是返回逻辑 0。如果选择了边沿和电平检测 （IRQMOD = 1），当 IRQ 引脚仍保持有效电平时，不能清除 IRQF。
1 IRQIE	**IRQ 中断使能** — 这个读 / 写控制位决定 IRQ 事件是否生成一个硬件中断请求。 0 关闭来自 IRQF 的硬件中断请求 （使用轮询）。 1 当 IRQF = 1 时请求硬件中断。
0 IRQMOD	**IRQ 检测模式** — 这个读 / 写控制位选择仅边沿检测还是边沿和电平检测。 IRQEDG 控制位决定边沿和电平的极性作为检测到中断请求事件的判断依据。如需了解更多信息，请参见 5.5.2.2 节 边沿和电平灵敏度 。 0 仅下降边沿或上升边沿的 IRQ 事件。 1 下降和低电平或上升边沿和高电平 IRQ 事件。

5.9.2 系统复位状态寄存器 (SRS)

这个寄存器的只读状态标记用于指示最近复位的源。将 1 写入 SBDFR 寄存器中的 BDFR 时，调试主机强制完成复位， SRS 中不设置任何状态位。在寄存器地址中写入任意值都会清除 COP 看门狗定时器，但不会对寄存器内容造成影响。这些位的复位状态取决于导致 MCU 复位的原因。

	7	6	5	4	3	2	1	0
R	POR	PIN	COP	ILOP	预留	ICG	LVD	0
W	写为任何值到 SIRMRS 地址，清除 COP 看门狗定时器。							
POR	1	0	0	0	0	0	1	0
LVR:	U	0	0	0	0	0	1	0
任何其它复位 :	0	(1)	(1)	(1)	0	(1)	0	0

U = 不受复位影响

[1] 复位时激活的任何复位源会造成设置相应的位；复位时未激活的源对应的位将被清除。

图 5-3. 系统复位状态 (SRS)

表 5-4. SRS 寄存器字段描述

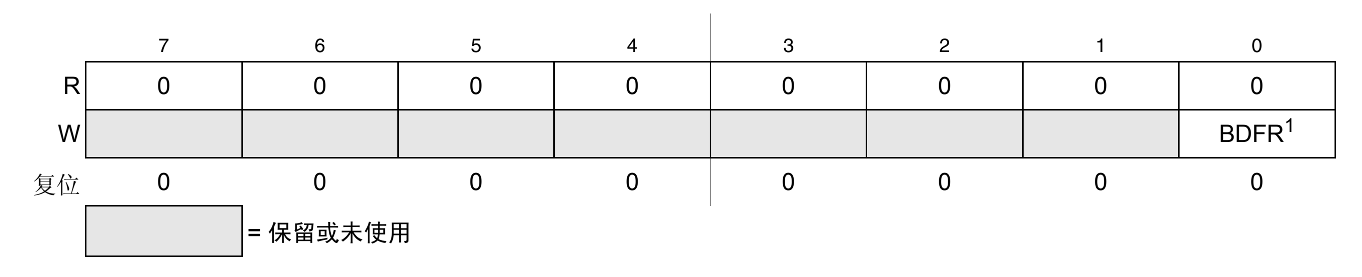
字段	描述
7 POR	**加电复位** — 复位由加电检测逻辑造成。由于此时内部电源电压在上升，所以还要设置低电压复位 （LVD）状态位，指示已发生复位，而内部电源低于 LVD 阈值。 . 0 非 POR 造成的复位。 1 POR 造成的复位。
6 PIN	**外部复位引脚** — 复位由外部复位引脚上的激活低电平造成。 0 非外部复位引脚造成的复位。 1 外部复位引脚造成的复位。

表 5-4. SRS 寄存器字段描述

字段	描述
5 COP	**计算机正常操作 (COP) 看门狗** — 复位由 COP 看门狗定时器超时导致。该复位源可以通过设置 COPE = 0 来阻拦。 0 非 COP 超时造成的复位。 1 COP 超时造成的复位。
4 LOP	**非法操作码** — 因试图执行未定义的或非法操作码导致复位。如果 SOPT 寄存器中 STOPE = 0，停止被关闭，此时， STOP 指令则被视为非法指令。如果 BDCSC 寄存器中 ENBDM = 0，关闭主动后台模式，BGND 指令则视为非法指令。 0 不是因非法操作码造成的复位。 1 非法操作码造成的复位。
2 ICG	**非法时钟生成模块复位** — ICG 模块复位造成复位。 0 不是由 ICG 模块造成的复位。 1 ICG 模块造成的复位。
1 LVD	**低电压检测** — 如果设置了 LVDRE 和 LVDSE 位，且电源降到 LVD 跳接电压以下，就会发生 LVD 复位。该位也可以由 POR 设置。 0 不是因 LVD 跳接电压或 POR 造成的复位。 1 LVD 跳接电压或 POR 造成的复位。

5.9.3 系统后台调试强制复位寄存器 (SBDFR)

这个寄存器只包括一个只写控制位。必须用串行后台命令，如 WRITE_BYTE 来写入 SBDFR。忽略从用户程序写入寄存器的尝试。读总是返回 0x00。



[1] 只能通过串行后台调试命令，不能通过用户程序写入 BDFR。

图 5-4. 系统后台调试强制复位寄存器 (SBDFR)

表 5-5. SBDFR 寄存器字段描述

字段	描述
0 BDFR	**后台调试强制复位** — 可以使用串行后台命令，如 WRITE_BYTE ，使外部调试主机对目标系统进行强制复位。在该位中写入逻辑 1 ，强制进行 MCU 复位。该位不能从用户程序中写入。

5.9.4 系统选项寄存器 (SOPT)

这个寄存器可随时读。 3 和 2 位未使用，总读为 0。这是单次写入有效的寄存器，因此只有复位后的第一次写入才有效。任何后续 SOPT 写入尝试 （有意或无意）都将被忽略，以避免对这些敏感的设置造成意外修改。 SOPT 应在用户复位初始化程序期间写入，设置期望的控制，即使期望的设置与复位设置相同。

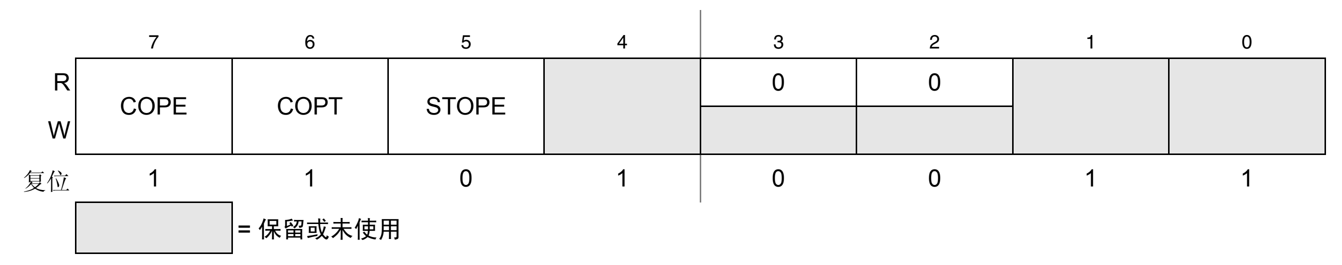


图 5-5. 系统选项寄存器 (SOPT)

表 5-6. SOPT 寄存器字段描述

字段	描述
7 COPE	**COP 看门狗使能** — 复位后，这个单次写入有效位缺省为 1。 0 COP 看门狗定时器关闭 1 COP 看门狗定时器使能 (超时时强制复位)。
6 COPT	**COP 看门狗超时** — 复位后，这个单次写入有效位缺省为 1。 0 选择短超时周期。 . 1 选择长超时周期。
5 STOPE	**停止模式使能** — 这个单次写入有效的位复位后，默认为 0，用来禁止停止模式。如果停止模式禁止期间，用户程序试图执行 STOP 指令，则会强制进行非法操作码复位。 0 停止模式禁止。 1 停止模式使能。

5.9.5 系统 MCLK 控制寄存器 (SMCLK)

这个寄存器用来控制 MCLK 时钟输出。

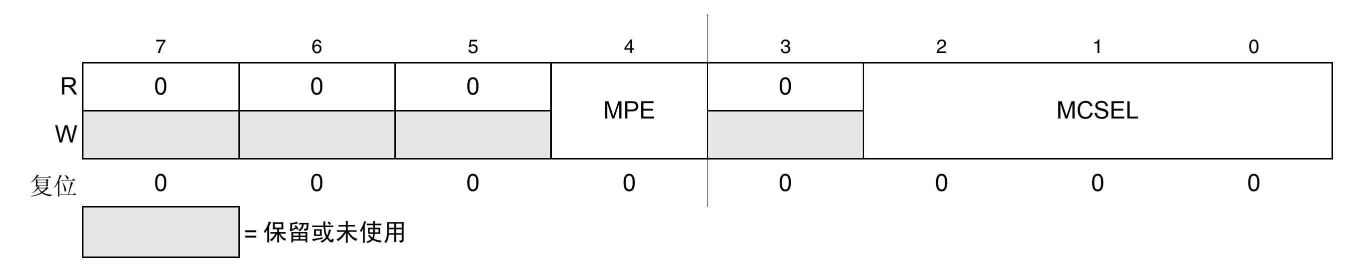


图 5-6. 系统 MCLK 控制寄存器 (SMCLK)

表 5-7. SMCLK 寄存器字段描述

字段	描述
4 MPE	**MCLK 引脚使能** — 这个位用来使能 MCLK 功能。 0 MCLK 输出关闭。 1 PTC2 引脚上 MCLK 输出使能。
2:0 MCSEL	**MCLK 除数选项** — 当 MCSEL 位不为全零时，这些位用来根据以下公式选择 MCLK 的除数。如果 MCSEL 位为全零，且 MPE 已设置，则该引脚驱向低电平位置。参见 公式 5-1。

MCLK 频率 = 总线时钟频率 ÷ (2 * MCSEL) *公式 5-1*

5.9.6 系统器件识别寄存器 (SDIDH, SDIDL)

这个只读寄存器允许主机开发系统确定 HCS08 的 MCU 类型和修订编号 。这样，开发软件可以识别特定的存储器块、寄存器和控制位在目标 MCU 中的位置。

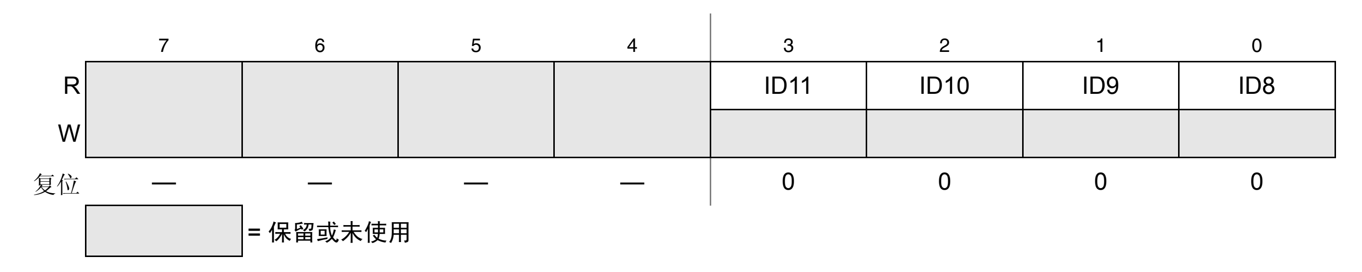


图 5-7. 系统器件识别寄存器 — 高 (SDIDH)

表 5-8. SDIDH 寄存器字段描述

字段	Description
7:4 预留	**7:4** 位预留。读这些位会导致不确定的值 ; 写入操作不产生影响。
3:0 ID[11:8]	部件标识号 — HCS08 系列中的每个部件都有唯一的标识号。系列采用硬编码方式，值为 0x001D。同时参见表 5-12 中的 ID 位。

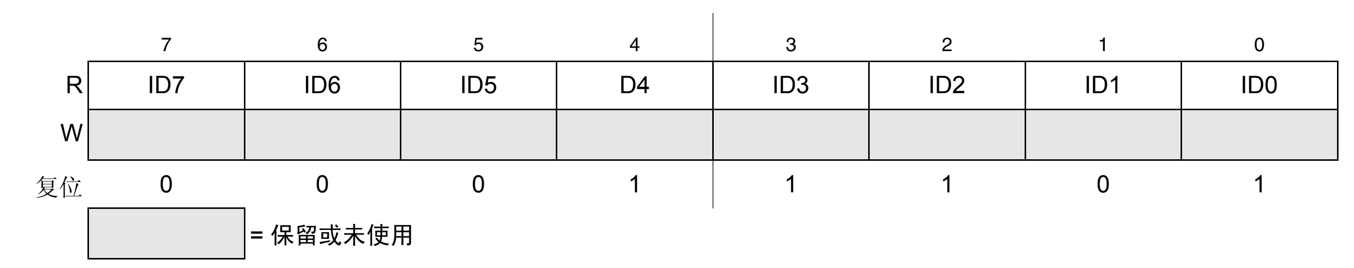


图 5-8. 系统器件识别寄存器 — 低 (SDIDL)

表 5-9. SDIDL 寄存器字段描述

字段	描述
7:0 ID[7:0]	部件标识号 — HCS08 系列中的每个部件都有唯一的标识号。系列采用硬编码方式，值为 0x001D。同时参见表 5-11 中的 ID 位。

5.9.7 系统实时中断状态和控制寄存器 (SRTISC)

这个寄存器包含一个只读状态标记，一个只写应答位，五个读 / 写时延选择和一个未使用的位，这个未使用的位总读为 0。

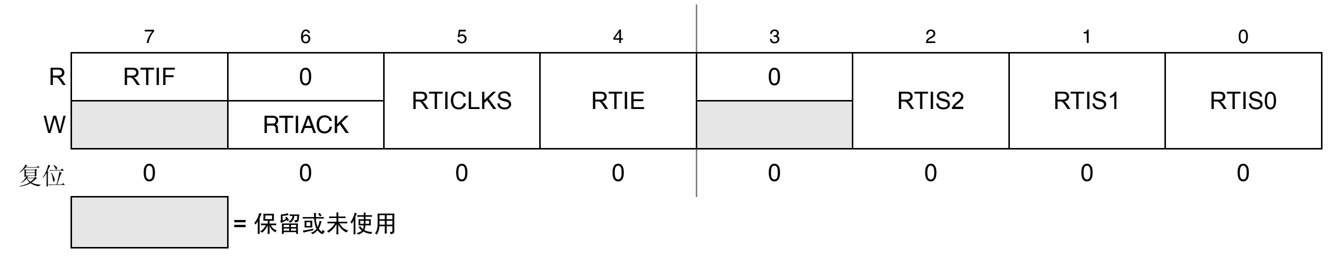


图 5-9. 系统 RTI 状态和控制寄存器 (SRTISC)

表 5-10. SRTISC 寄存器字段描述

字段	描述
7 RTIF	**时实中断标志** — 这个只读状态位指示定期唤醒定时器已超时。 0 定期唤醒定时器未超时。 1 定期唤醒定时器超时。
6 RTIACK	**实时中断应答** — 这个单次写入有效位用来应答实时中断请求 (写入 1，清除 RTIF)。写入 0 无意义或无影响。读操作总返回逻辑 0。
5 RTICLKS	**实时中断时钟选择** — 这个读 / 写位选择实时中断的时钟源。 0 实时中断请求时钟源是内部 1 kHz 振荡器。 1 实时中断请求时钟源是外部时钟。
4 RTIE	**实时中断使能** — 这个读写位使能实时中断。 0 实时中断关闭。 1 实时中断使能。
2:0 RTIS[2:0]	**实时中断时延选择** — 这些读 / 写位选择 RTI 的唤醒时延。实时中断的时钟源是自时钟源，以 1 kHz 频率振荡，独立于其它 MCU 时钟源。采用外部时钟源，时延将为晶振频率除以 RTIS2:RTIS1:RTIS0 中的值。参见表 5-11。

表 5-11. 实时中断频率

RTIS2:RTIS1:RTIS0	1 kHz 时钟源时延[1]	采用外部时钟源时延 (晶振频率)
0:0:0	关闭定时唤醒定时器	关闭定时唤醒定时器
0:0:1	8 ms	除以 256
0:1:0	32 ms	除以 1024
0:1:1	64 ms	除以 2048
1:0:0	128 ms	除以 4096
1:0:1	256 ms	除以 8192
1:1:0	512 ms	除以 16384
1:1:1	1.024 s	除以 32768

[1] 栏中显示的正常值基于 f_{RTI} = 1 kHz。参见附录 A 电气特性和时序规范中的 f_{RTI}, 了解这些值的容限。

5.9.8 系统电源管理状态和控制寄存器 1 (SPMSC1)

	7	6	5	4	3	2	1[1]	0
R	LVDF	0	LVDIE	LVDRE[2]	LVDSE[2]	LVDE[2]		BGBE
W		LVDACK						
复位	0	0	0	1	1	1	0	0

= 保留或未使用

[1] Bit 1 为预留位，必须写为 0。

[2] 复位后，这个位只能写入一次。其它写入操作被忽略。

图 5-10. 系统电源管理状态和控制寄存器 1 (SPMSC1)

表 5-12. SPMSC1 寄存器字段描述

字段	描述
7 LVDF	低压检测标志 — 如果 LVDE = 1, 这个只读状态位指示低压检测事件。
6 LVDACK	低压检测应答 — 这个只写位用来应答低压检测错误 （写 1，清除 LVDF）。读操作总返回 0。
5 LVDIE	低压检测中断使能 — 这个读 / 写位使能 LVDF 的硬件中断请求。 0 关闭硬件中断 (采用轮询)。 1 当 LVDF = 1，请求硬件中断。
4 LVDRE	低压检测复位使能 — 这个读 / 写位支持 LVDF 事件生成硬件复位 （假设 LVDE = 1）。 0 LVD 事件不生成硬件复位。 1 当 LVDF = 1 时，强制 MCU 复位。
3 LVDSE	低压检测停止使能 — 如果 LVDE = 1，这个读 / 写位决定当 MCU 处于停止模式时是否低压检测。 0 停止模式期间低压检测关闭。 1 停止模式期间低压检测使能。
2 LVDE	低压检测使能 — 这个读 / 写位支持低压检测逻辑，并且限定该寄存器中的其他位的操作。 0 LVD 逻辑关闭。 1 LVD 逻辑使能。
0 BGBE	带隙缓冲器使能 — BGBE 位使能内部缓冲器，提供带隙电压参考，可供 ADC 模块在其一个内部通路上使用。 0 带隙缓冲器关闭。 1 带隙缓冲器使能。

5.9.9 系统电源管理状态和控制寄存器 2 (SPMSC2)

该寄存器用来报告低电压告警的状态，配置 MCU 的停止模式行为。

	7	6	5	4	3	2	1	0
R	LVWF	0	LVDV[1]	LVWV	PPDF	0		PPDC[2]
W		LVWACK				PPDACK		
加电复位	0[3]	0	0	0	0	0	0	0
LVD 复位	0[2]	0	U	U	0	0	0	0
任何其它复位 :	0[2]	0	U	U	0	0	0	0

= 保留或未使用　　U = 不受复位影响

1 POR 后这个位只能写入一次，后续写入均忽略。
2 复位后，这个位只能写入一次，后续写入均忽略。
3 当 V_{Supply} 转换低于跳变点或者复位后 V_{Supply} 已经低于 V_{LVW} 时，置位 LVWF。

图 5-11. 系统电源管理状态和控制寄存器 2 (SPMSC2)

表 5-13. SPMSC2 寄存器字段描述

字段	描述
7 LVWF	**低压告警标志** — LVWF 位指示低压警告状态。 0 低压告警未出现。 1 低压告警已出现或出现过。
6 LVWACK	**低压告警应答** — LVWACK 位是低压告警应答 。 如果没有出现低压告警，写 1 到 LVWACK，将清 0 LVWF。
5 LVDV	**低压检测电压选择** — LVDV 位选择 LVD 跳变点电压 (V_{LVD})。 0 选择低跳变点 ($V_{LVD} = V_{LVDL}$)。 1 选择高跳变点 ($V_{LVD} = V_{LVDH}$)。
4 LVWV	**低压告警电压选择** — LVWV 位选择 LVW 跳变点电压 (V_{LVW})。 0 选择低跳变点 ($V_{LVW} = V_{LVWL}$)。 1 选择高跳变点 ($V_{LVW} = V_{LVWH}$)。
3 PPDF	**局部功率下降标志** — PPDF 位标识 MCU 已退出停止 2 模式。 0 非停止 2 模式恢复。 1 停止 2 模式恢复。
2 PPDACK	**局部功率下降应答** — 将 PPDACK 位写为 1，清除 PPDF 位。
0 PPDC	**局部功率下降控制** — 这个单次写入有效的 PPDC 位控制选择停上 2 还是停止 3 模式。 0 使能停止 3 模式。 1 停止 2，部分功率下降，模式使能 。

5.9.10 系统选项寄存器 2 (SOPT2)

这个高页寄存器包含配置 MC9S08AC60 系列特定功能的位。

	7	6	5	4	3	2	1	0
R	COPCLKS[1]	0	0	0	TPMCCFG	0	0	0
W								
复位	1	0	0	0	1	0	0	0

= 保留或未使用

1 复位后，这个位只能写入一次。后续写入将忽略。

图 5-12. 系统选项寄存器 2 (SOPT2)

表 5-14. SOPT2 寄存器字段描述

字段	描述
7 COPCLKS	**COP 看门狗时钟选择** — 这个单次写入有效的位选择 COP 看门狗时钟源。 0 采用内部 1 kHz 时钟作为 COP 的时钟源。 1 采用总线时钟作为 COP 的时钟源。
3 TPMCCFG	**TPM 时钟配置** — 配置定时器 / 脉宽调制器时钟信号。 0 通过 IRQ 引脚，提供 TPMCLK 给 TPM1、TPM2 和 TPM3，不提供 TPMCLK1 和 TPMCLK2。 1 分别提供 TPM1CLK、 TPM2CLK 和 TPMCLK 给 TPM1、TPM2 和 TPM3。

第 6 章
并行输入 / 输出

6.1 介绍

本章介绍与并行输入 / 输出 （I/O）相关的软件控制。MC9S08AC60 系列有 7 个 I/O 端口，总共包含 54 个通用 I/O 引脚。有关这些引脚的逻辑和硬件方面的更多信息，请参见 第 2 章 引脚和连接 。

这些引脚大多和片上外围设备，如定时器系统、通信系统和键盘中断等复用。当这些模块没有控制端口引脚时，它们回复到通用 I/O 控制。

注意

不是所有封装都具有全部通用输入 / 输出引脚。为了避免从输入浮空引脚消耗过多电流，应用程序中的用户复位初始化程序必须要么使能片上上拉器件，要么将未连接引脚的方向更改为输出，使引脚不会浮空。

6.2 引脚描述

MC9S08AC60 系列在七个端口 (PTA 到 PTG) 提供 54 个并行 I/O 引脚。不是所有封装都具有全部引脚。各种封装提供的并行 I/O 引脚的信息，请参见 第 2 章 引脚和连接。所有这些引脚当没有被其他片上外设系统使用时可用作通用输入输出。

复位后，复用的外围功能关闭，这样这些引脚被并行 I/O 控制。所有并行 I/O 配置为输入 （PTxDDn = 0）。每个引脚的引脚控制功能配置如下：斜率控制使能 （PTxSEn = 1）、选择低驱动强度 （PTxDSn = 0）、内部上拉关闭 （PTxPEn = 0）。

6.3 并行 I/O 控制

通过端口数据寄存器读 / 写并行 I/O。输入输出方向由端口数据方向寄存器控制。下面的结构图举例了一个引脚的并行 I/O 端口功能。

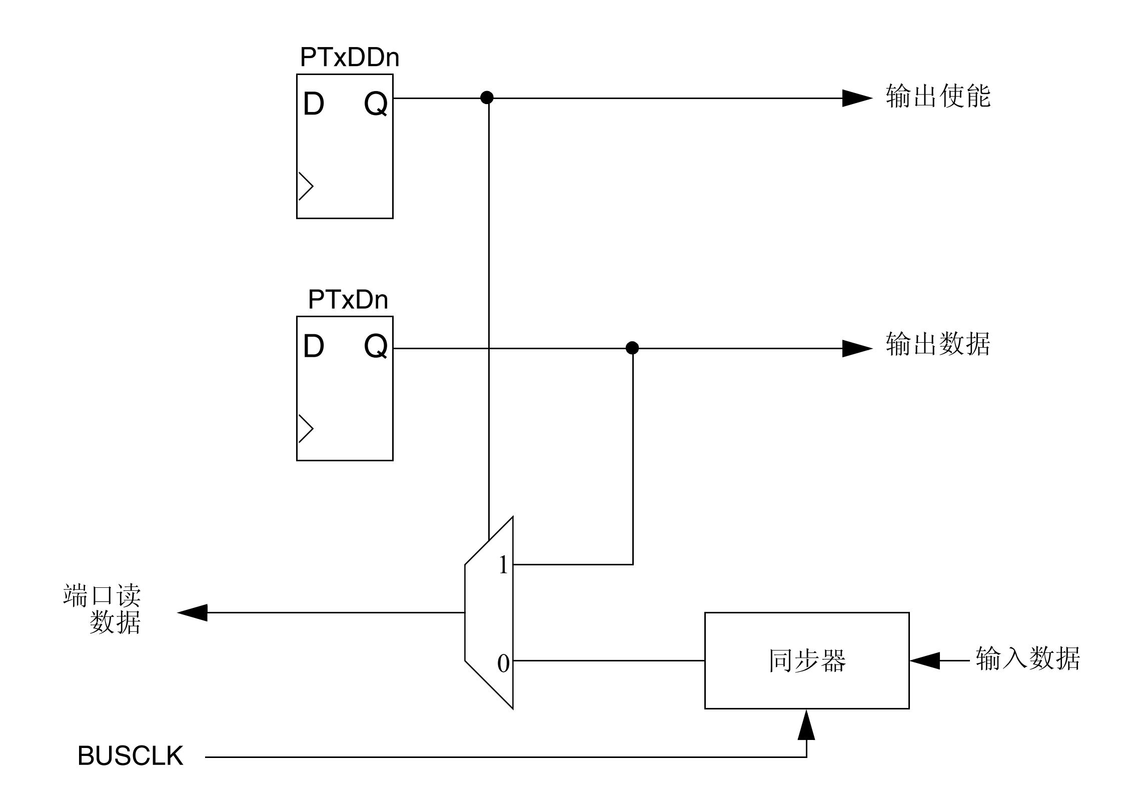


图 6-1. 并行 I/O 结构图

数据方向控制位决定是否使能引脚输出驱动。它们还控制端口数据寄存器读什么。每个端口引脚都有一个数据方向寄存器位。PTxDDn = 0 时，相应的引脚是输入，PTxD 读操作返回引脚值。PTxDDn = 1 时，相应的引脚为输出，PTxD 的读操作返回最后一次写入到端口数据寄存器的值 。外围模块或系统功能控制一个端口引脚时，数据方向寄存器位仍然控制读端口数据寄存器时返回的内容，即使外围系统控制实际的引脚方向。

当一个引脚的复用模拟功能使能时，所有数字引脚功能均关闭。对于启用了共享模拟功能的任何位，读端口数据寄存器返回 0。一般而言，一个引脚若复用数字和模拟功能，模拟功能优先，这样，如果数字和模拟都使能时，模拟功能控制这个引脚。

建议在改变端口引脚方向成为输出之前写端口数据寄存器。这确保引脚不会因为旧的数据值碰巧在端口数据寄存器中而临时驱动。

6.4 引脚控制

引脚控制寄存器位于存储器的高位页面寄存器块中。这些寄存器用来控制 I/O 引脚的上拉、斜率和驱动强度。这个引脚控制寄存器的操作独立于并行 I/O 寄存器。

6.4.1 内部上拉使能

在一个上拉使能寄存器 (PTxPEn) 中设置相应的位，可以使能内部上拉器件。如果并行 I/O 控制逻辑或任何复用的外围功能配置该引脚为输出，上拉器件关闭，而不管相应的上拉使能寄存器位的状态。如果引脚由模拟功能控制，上拉器件也关闭。

6.4.2 输出斜率控制使能

在其中一个斜率控制寄存器 (PTxSEn) 中设置相应的位，可以使能斜率控制。使能后，斜率控制限制输出可以转变的速率，这样可以减少 EMC 辐射。斜率控制不影响配置为输入的引脚。

6.4.3 输出驱动强度选择

在一个驱动强度选择寄存器 (PTxDSn) 中设置相应的位，可以选择一个输出引脚，拥有高输出驱动强度。选择高驱动强度时，引脚可以拉和灌更大的电流。即使每个 I/O 引脚可以选择为高驱，用户也必须保证不超过芯片的总拉电流和灌电流限制。驱动强度选择会影响 I/O 引脚的 DC 行为，但是，也会影响 AC 操作。高驱允许引脚以与低驱引脚相同的交换速度，将更大的载荷变为更小的载荷。正是因为如此，将引脚使能为高驱，可能会影响 EMC 辐射。

6.5 停止模式中的引脚行为

根据进入的不同停止模式，执行 STOP 指令， I/O 会有不同的功能。下面阐述了不同停止模式的引脚行为：

- 停止 2 模式是局部断电模式，I/O 寄存器保持执行 STOP 指令前的状态。在执行 STOP 指令，使 MCU 进入停止 2 模式前，应将 CPU 寄存器状态和 I/O 寄存器状态保存在 RAM 中。从停止 2 模式恢复时，用户访问任何 I/O 前，都应检查 SPMSC2 寄存器里的 PPDF 位状态。如果 PPDF 位为 0，I/O 必须初始化，如同发生过上电复位那样。如果 PPDF 位为 1， I/O 数据之前存储在 RAM 中，在执行 STOP 指令前，外围设备可能要求初始化，并恢复到停止前的情况。然后用户必须向 SPMSC2 寄存器中的 PPDACK 位写入 1。现在，再次允许用户应用程序访问 I/O。
- 停止 3 模式下，所有 I/O 都保留，因为内部逻辑电路处于通电状态。恢复后，用户可以使用正常的 I/O 功能。

6.6 并行 I/O 和引脚控制寄存器

本小节介绍了与并行 I/O 端口和引脚控制功能相关的寄存器信息。这些并行 I/O 寄存器位于存储器映射的 0 页面中。引脚控制寄存器位于存储器寄存器段的高位页面。

请参阅 第 4 章 存储器中的表格，了解 I/O 及其引脚控制寄存器的绝对地址分配信息。本小节只提到寄存器和控制位的名称，请参阅飞思卡尔提供的等同或头文件，将这些名称转换为相应的绝对地址。

6.6.1 A 端口 I/O 寄存器 (PTAD 和 PTADD)

A 端口并行 I/O 功能受下列的寄存器控制。

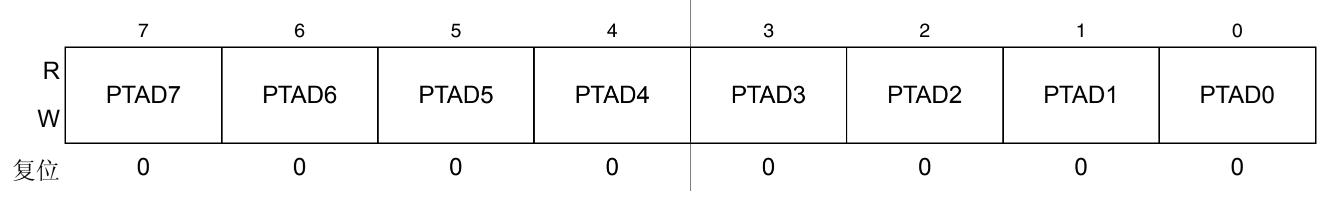


图 6-2. A 端口数据寄存器 (PTAD)

表 6-1. PTAD 寄存器字段描述

字段	描述
7:0 PTADn	**A 端口数据寄存器位** — 对于配置为输入的 A 端口引脚，读数返回引脚上的逻辑电平。对于配置为输出的 A 端口引脚，读数返回最后一次写入寄存器的值。 写入值被锁存在本寄存器的所有位中。对于配置为输出的 A 端口引脚，逻辑电平驱动相应的 MCU 引脚。 复位强制 PTAD 都为 0，但是这些 0 未被驱出相应的引脚，因为复位也会将所有端口引脚配置为上拉关闭的高抗阻输入。

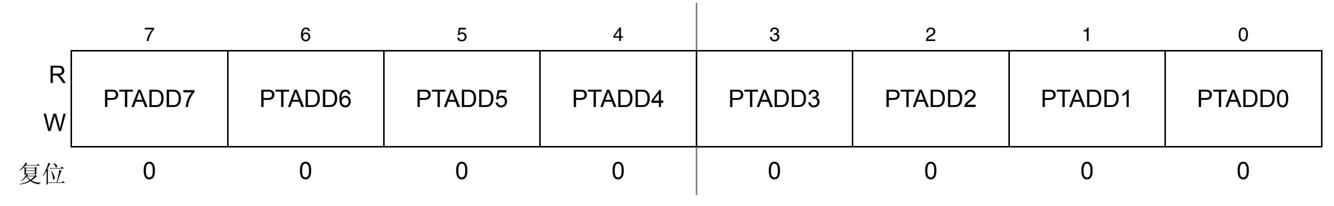


图 6-3. A 端口数据方向寄存器 (PTADD)

表 6-2. PTADD 寄存器字段描述

字段	描述
7, 2:0 PTADDn	**A 端口数据方向位** — 这些读 / 写位控制着 A 端口引脚的方向以及 PTAD 读取的内容。 0 输入 （输出驱动关闭），读数返回引脚值。 1 A 端口位 n 输出驱动使能，PTAD 读数返回 PTADn 内容。

6.6.2 端口 A 引脚控制寄存器 (PTAPE, PTASE, PTADS)

除了 I/O 控制外，端口 A 引脚还受下列寄存器控制。

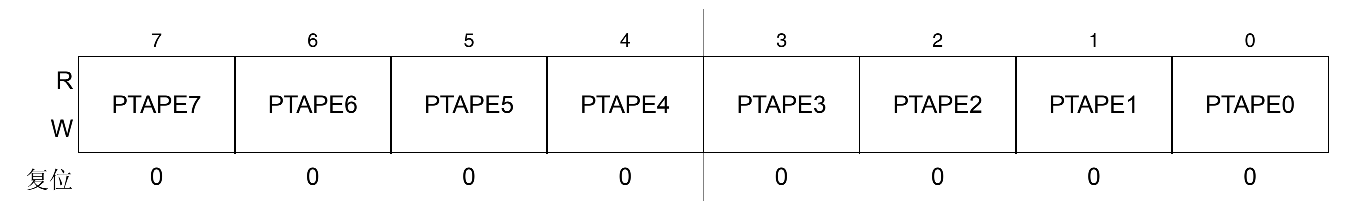


图 6-4. A 端口内部上拉使能寄存器 (PTAPE)

表 6-3. PTAPE 寄存器字段描述

字段	描述
7:0 PTAPEn	**A 端口内部上拉使能位** — 对于 PTA 引脚，这些控制位决定相关 PTA 引脚的内部上拉器件是否使能。对于配置为输出的 A 端口引脚，这些位不会产生影响，同时内部上拉器件关闭。 0 A 端口位 n 内部上拉器件关闭。 1 A 端口位 n 内部上拉器件使能。

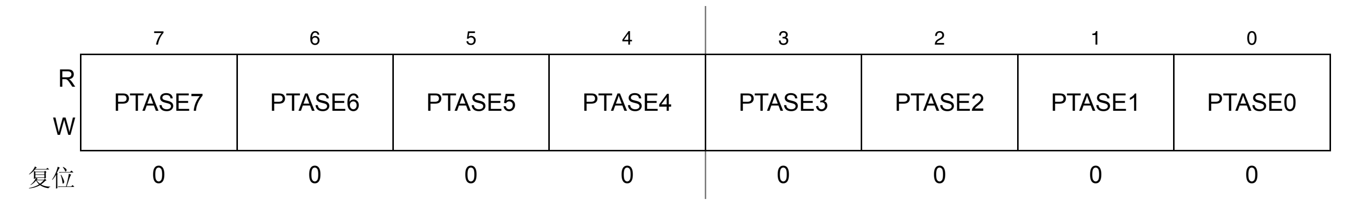


图 6-5. A 端口斜率控制使能寄存器 (PTASE)

表 6-4. PTASE 寄存器字段描述

字段	描述
7:0 PTASEn	**A 端口输出斜率使能位** — 这些控制位决定是否为相关 PTA 引脚使能输出斜率控制。对于配置为输入的 A 端口引脚，这些位不会产生任何影响。 0 A 端口位 n 输出斜率控制关闭。 1 A 端口位 n 输出斜率控制使能。

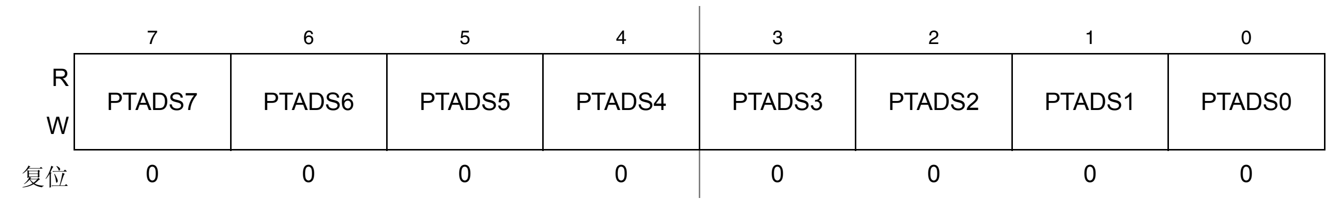


图 6-6. A 端口驱动强度选择寄存器 (PTADS)

表 6-5. PTADS 寄存器字段描述

字段	描述
7:0 PTADSn	**A 端口输出驱动强度选择位** — 这些控制位为 PTA 引脚设置低 / 高输出驱动。 0 A 端口位 n 低输出驱动使能。 1 A 端口位 n 高输出驱动使能。

6.6.3 B 端口 I/O 寄存器 (PTBD 和 PTBDD)

B 端口并行 I/O 功能由下列寄存器控制。

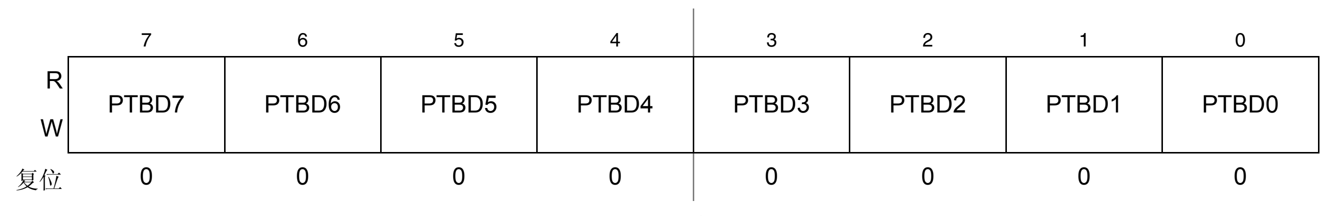


图 6-7. B 端口数据寄存器 (PTBD)

表 6-6. PTBD 寄存器字段描述

字段	描述
7:0 PTBD[7:0]	**B 端口数据寄存器位** — 对于配置为输入的 B 端口引脚，读数返回引脚上的逻辑电平。对于配置为输出的 B 端口引脚，读数返回最后一次写入寄存器的值。 写入值被锁存在本寄存器的所有位中。对于配置为输出的 B 端口引脚，逻辑电平驱动相应的 MCU 引脚。 复位强制 PTBD 都为 0，但是这些 0 未被驱出相应的引脚，因为复位也会将所有端口引脚配置为上拉关闭的高抗阻输入。

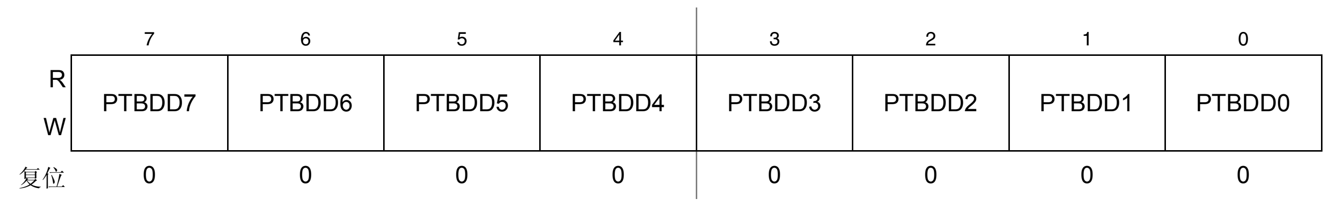


图 6-8. B 端口数据方向寄存器 (PTBDD)

表 6-7. PTBDD 寄存器字段描述

字段	描述
7:0 PTBDD[7:0]	**B 端口数据方向位** — 这些读 / 写位控制着 B 端口引脚的方向以及 PTBD 读取的内容。 0 输入 （输出驱动关闭），读数返回引脚值。 1 B 端口位 n 输出驱动使能，PTBD 读数返回 PTBDn 内容。

6.6.4 B 端口引脚 控制寄存器 (PTBPE, PTBSE, PTBDS)

除了 I/O 控制外，B 端口引脚还由下列寄存器控制。

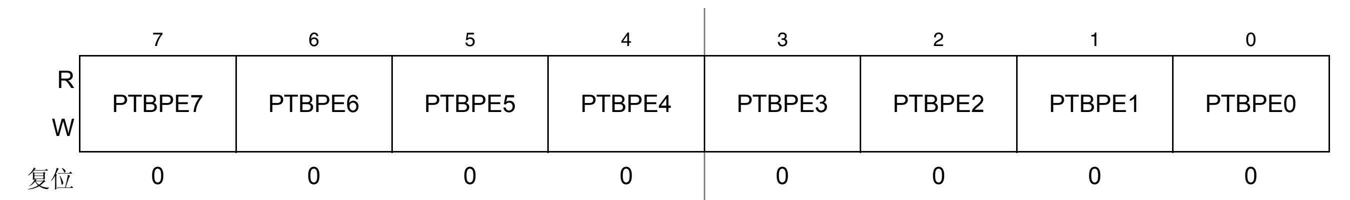


图 6-9. B 端口内部上拉使能寄存器 (PTBPE)

表 6-8. PTBPE 寄存器字段描述

字段	描述
7:0 PTBPE[7:0]	**B 端口内部上拉使能位** — 对于 PTB 引脚，这些控制位决定相关 PTB 引脚的内部上拉器件是否使能。对于配置为输出的 B 端口引脚，这些位不会产生影响，同时内部上拉器件关闭。 0 B 端口位 n 内部上拉器件关闭。 1 B 端口位 n 内部上拉器件使能。

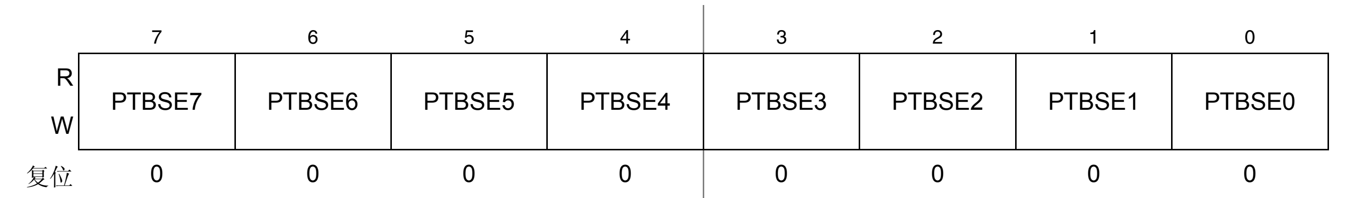


图 6-10. B 端口斜率控制使能寄存器 (PTBSE)

表 6-9. PTBSE 寄存器字段描述

字段	描述
7:0 PTBSE[7:0]	**B 端口输出斜率使能位** — 这些控制位决定是否为相关 PTB 引脚使能输出斜率控制。对于配置为输入的 B 端口引脚，这些位不会产生任何影响。 0 B 端口位 n 输出斜率控制关闭。 1 B 端口位 n 输出斜率控制使能。

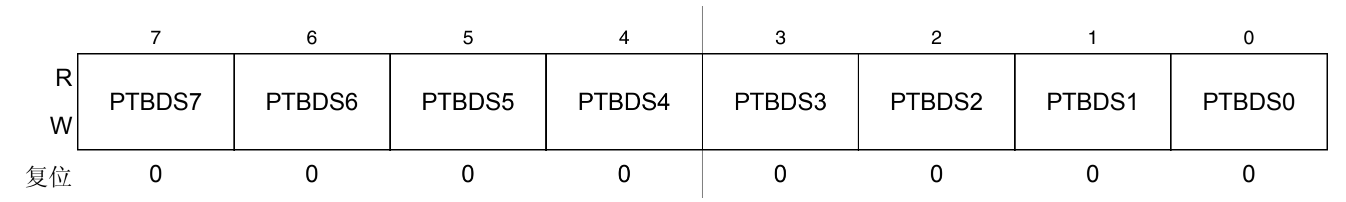


图 6-11. B 端口驱动强度选择寄存器 (PTBDS)

表 6-10. PTBDS 寄存器字段描述

字段	描述
7:0 PTBDS[7:0]	**B 端口输出驱动强度选择位** — 这些控制位为 PTB 引脚设置低 / 高输出驱动。 0 B 端口位 n 低输出驱动使能。 1 B 端口位 n 高输出驱动使能。

6.6.5 C 端口 I/O 寄存器 (PTCD 和 PTCDD)

C 端口并行 I/O 功能受下列的寄存器控制。

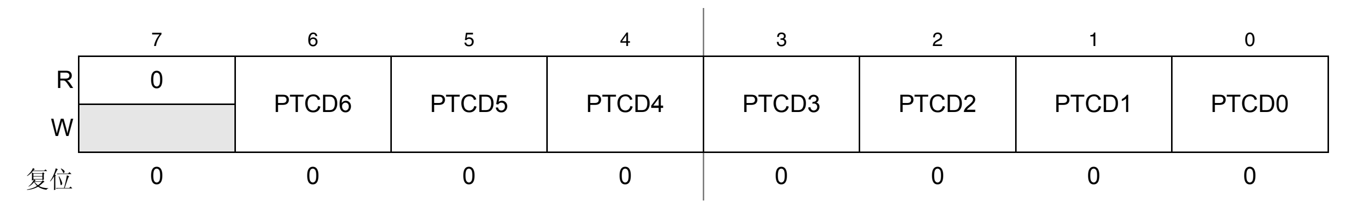


图 6-12. C 端口数据寄存器 (PTCD)

表 6-11. PTCD 寄存器字段描述

字段	描述
6:0 PTCD[6:0]	**C 端口数据寄存器位** — 对于配置为输入的 C 端口引脚，读数返回引脚上的逻辑电平。对于配置为输出的 C 端口引脚，读数返回最后一次写入寄存器的值。 写入值被锁存在本寄存器的所有位中。对于配置为输出的 C 端口引脚，逻辑电平驱动相应的 MCU 引脚。 复位强制 PTCD 都为 0，但是这些 0 未被驱出相应的引脚，因为复位也会将所有端口引脚配置为上拉关闭的高抗阻输入。

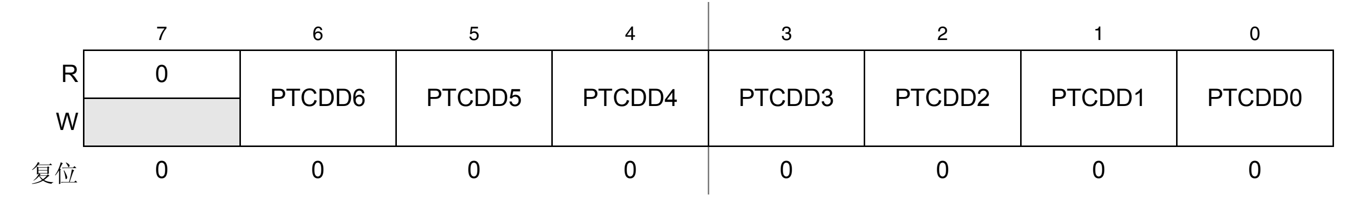


图 6-13. C 端口数据方向寄存器 (PTCDD)

表 6-12. PTCDD 寄存器字段描述

字段	描述
6:0 PTCDD[6:0]	**C 端口数据方向位** — 这些读 / 写位控制着 C 端口引脚的方向以及 PTCD 读取的内容。 0 输入 （输出驱动关闭），读数返回引脚值。 1 C 端口位 n 输出驱动使能，PTCD 读数返回 PTCDn 内容。

6.6.6 C 端口引脚控制寄存器 (PTCPE, PTCSE, PTCDS)

除了 I/O 控制外，C 端口引脚还由下列寄存器控制。

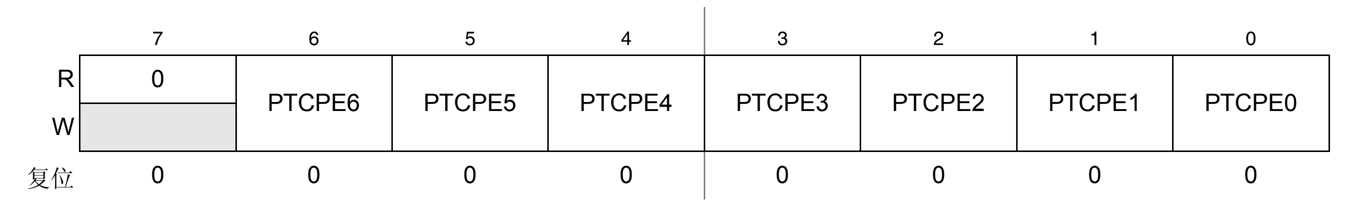


图 6-14. C 端口内部上拉使能寄存器 (PTCPE)

表 6-13. PTCPE 寄存器字段描述

字段	Description
6:0 PTCPE[6:0]	**C 端口内部上拉使能位** — 对于 PTC 引脚，这些控制位决定相关 PTC 引脚的内部上拉器件是否使能。对于配置为输出的 C 端口引脚，这些位不会产生影响，同时内部上拉器件关闭。 0 C 端口位 n 内部上拉器件关闭。 1 C 端口位 n 内部上拉器件使能。

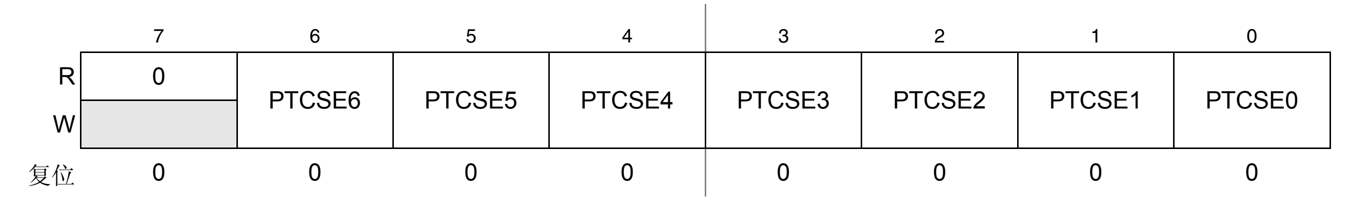


图 6-15. C 端口斜率控制使能寄存器 (PTCSE)

表 6-14. PTCSE 寄存器字段描述

字段	描述
6:0 PTCSE[6:0]	**C 端口输出斜率使能位** — 这些控制位决定是否为相关 PTC 引脚使能输出斜率控制。对于配置为输入的 C 端口引脚，这些位不会产生任何影响。 0 C 端口位 n 输出斜率控制关闭。 1 C 端口位 n 输出斜率控制使能。

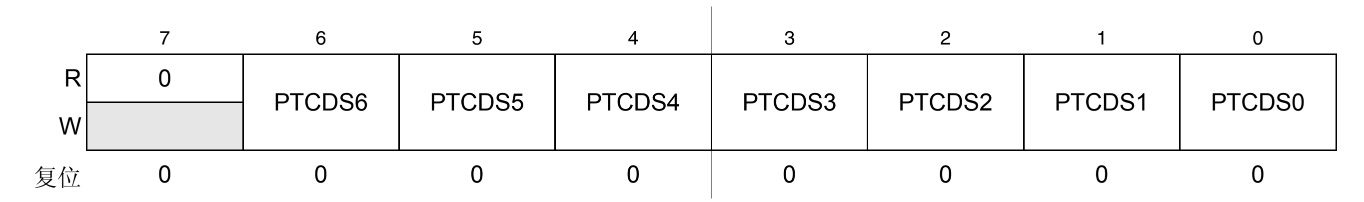


图 6-16. C 端口驱动强度选择寄存器 (PTCDS)

表 6-15. PTCDS 寄存器字段描述

字段	描述
6:0 PTCDS[6:0]	**C 端口输出驱动强度选择位** — 这些控制位为 PTC 引脚设置低 / 高输出驱动。 0 C 端口位 n 低输出驱动使能。 1 C 端口位 n 高输出驱动使能。

6.6.7 D 端口 I/O 寄存器 (PTDD 和 PTDDD)

D 端口并行 I/O 功能由下列寄存器控制。

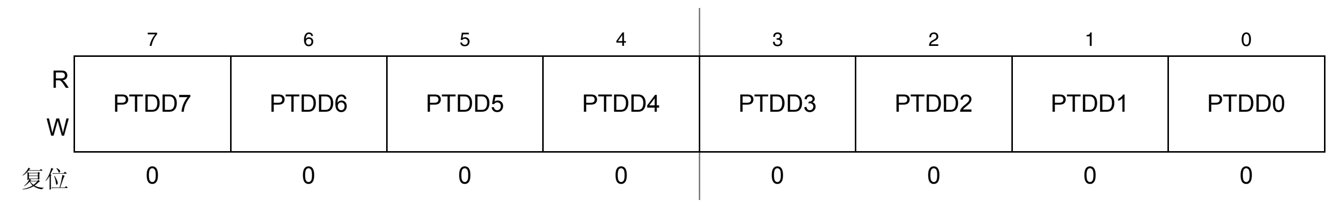


图 6-17. D 端口数据寄存器 (PTDD)

表 6-16. PTDD 寄存器字段描述

字段	描述
7:0 PTDD[7:0]	**D 端口数据寄存器位** — 对于配置为输入的 D 端口引脚，读数返回引脚上的逻辑电平。对于配置为输出的 D 端口引脚，读数返回最后一次写入寄存器的值。 写入值被锁存在本寄存器的所有位中。对于配置为输出的 D 端口引脚，逻辑电平驱动相应的 MCU 引脚。 复位强制 PTDD 都为 0，但是这些 0 未被驱出相应的引脚，因为复位也会将所有端口引脚配置为上拉关闭的高抗阻输入。

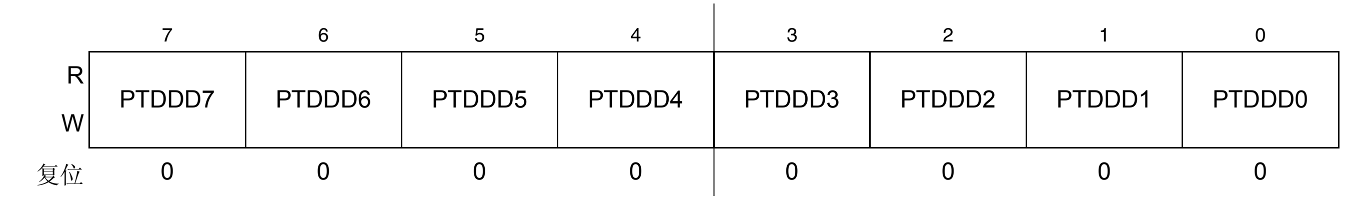


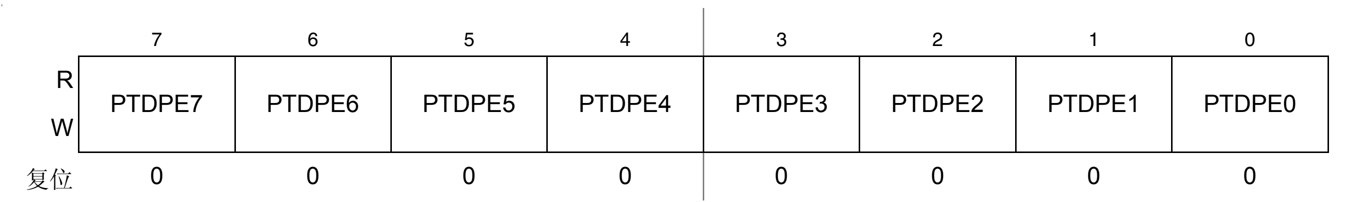
图 6-18. D 端口数据方向寄存器 (PTDDD)

表 6-17. PTDDD 寄存器字段描述

字段	描述
7:0 PTDDD[7:0]	**D 端口数据方向位** — 这些读 / 写位控制着 D 端口引脚的方向以及 PTDD 读取的内容。 0 输入 （输出驱动关闭），读数返回引脚值。 1 D 端口位 n 输出驱动使能，PTDD 读数返回 PTDDn 内容。

6.6.8 D 端口引脚 控制 寄存器 (PTDPE, PTDSE, PTDDS)

除了 I/O 控制外，D 端口引脚还受下列寄存器控制。

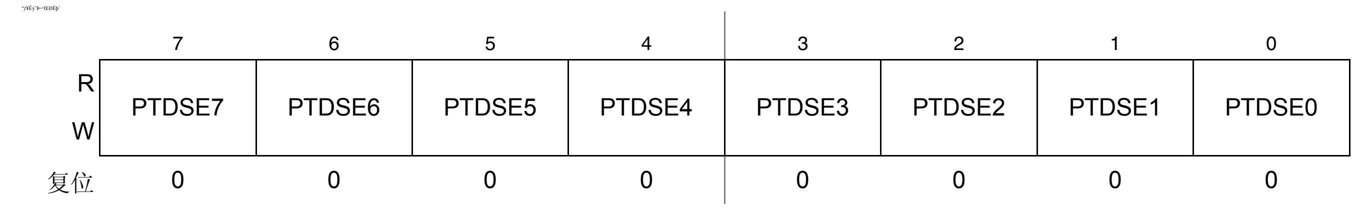


	7	6	5	4	3	2	1	0
R W	PTDPE7	PTDPE6	PTDPE5	PTDPE4	PTDPE3	PTDPE2	PTDPE1	PTDPE0
复位	0	0	0	0	0	0	0	0

图 6-19. D 端口内部上拉使能寄存器 (PTDPE)

表 6-18. PTDPE 寄存器字段描述

字段	描述
7:0 PTDPE[7:0]	**D 端口内部上拉使能位** — 对于 PTD 引脚，这些控制位决定相关 PTD 引脚的内部上拉器件是否使能。对于配置为输出的 D 端口引脚，这些位不会产生影响，同时内部上拉器件关闭。 0 D 端口位 n 内部上拉器件关闭。 1 D 端口位 n 内部上拉器件使能。

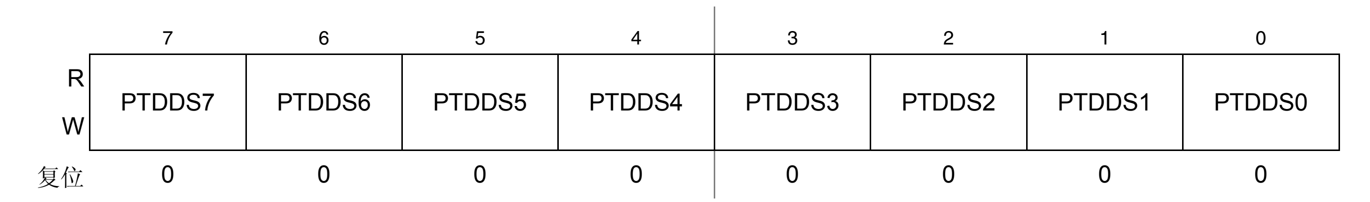


	7	6	5	4	3	2	1	0
R W	PTDSE7	PTDSE6	PTDSE5	PTDSE4	PTDSE3	PTDSE2	PTDSE1	PTDSE0
复位	0	0	0	0	0	0	0	0

图 6-20. D 端口斜率控制使能寄存器 (PTDSE)

表 6-19. PTDSE 寄存器字段描述

字段	描述
7:0 PTDSE[7:0]	**D 端口输出斜率使能位** — 这些控制位决定是否为相关 PTD 引脚使能输出斜率控制。对于配置为输入的 D 端口引脚，这些位不会产生任何影响。 0 D 端口位 n 输出斜率控制关闭。 1 D 端口位 n 输出斜率控制使能。



	7	6	5	4	3	2	1	0
R W	PTDDS7	PTDDS6	PTDDS5	PTDDS4	PTDDS3	PTDDS2	PTDDS1	PTDDS0
复位	0	0	0	0	0	0	0	0

图 6-21. D 端口驱动强度选择寄存器 (PTDDS)

表 6-20. PTDDS 寄存器字段描述

字段	描述
7:0 PTDDS[7:0]	**D 端口输出驱动强度选择位** — 这些控制位为 PTD 引脚设置低 / 高输出驱动。 0 D 端口位 n 低输出驱动使能。 1 D 端口位 n 高输出驱动使能。

6.6.9 端口 E I/O 寄存器 (PTED 和 PTEDD)

端口 E 并行 I/O 功能受下列寄存器控制。

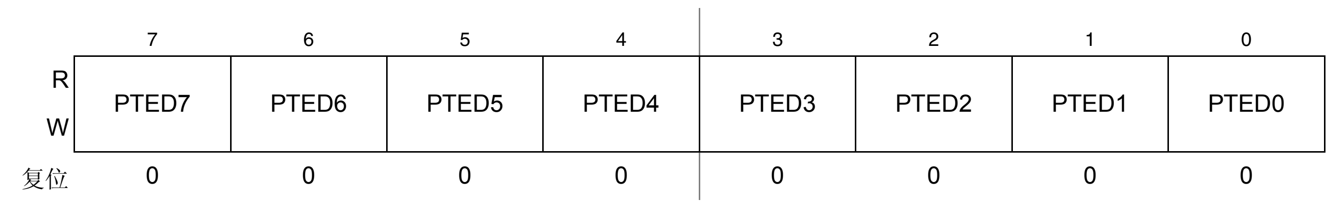


图 6-22. E 端口数据寄存器 (PTED)

表 6-21. PTED 寄存器字段描述

字段	描述
7:0 PTED[7:0]	**E 端口数据寄存器位** — 对于配置为输入的 E 端口引脚，读数返回引脚上的逻辑电平。对于配置为输出的 E 端口引脚，读数返回最后一次写入寄存器的值。 写入值被锁存在本寄存器的所有位中。对于配置为输出的 E 端口引脚，逻辑电平驱动相应的 MCU 引脚。 复位强制 PTED 都为 0，但是这些 0 未被驱出相应的引脚，因为复位也会将所有端口引脚配置为上拉关闭的高抗阻输入。

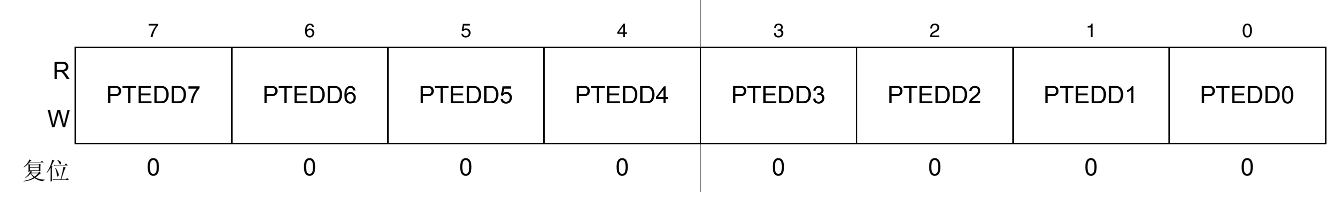


图 6-23. E 端口数据方向寄存器 (PTEDD)

表 6-22. PTEDD 寄存器字段描述

字段	描述
7:0 PTEDD[7:0]	**E 端口数据方向位** — 这些读 / 写位控制着 E 端口引脚的方向以及 PTED 读取的内容。 0 输入 （输出驱动关闭），读数返回引脚值。 1 E 端口位 n 输出驱动使能，PTED 读数返回 PTEDn 内容。

6.6.10 E 端口引脚控制寄存器 (PTEPE, PTESE, PTEDS)

除了 I/O 控制外，E 端口引脚还受下列寄存器控制。

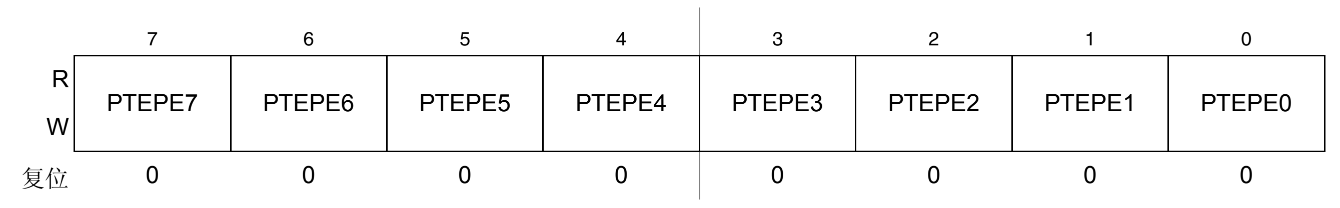


图 6-24. E 端口内部上拉使能寄存器 (PTEPE)

表 6-23. PTEPE 寄存器字段描述

字段	描述
7:0 PTEPE[7:0]	**E 端口内部上拉使能位** — 对于 PTE 引脚，这些控制位决定相关 PTE 引脚的内部上拉器件是否使能。对于配置为输出的 E 端口引脚，这些位不会产生影响，同时内部上拉器件关闭。 0 E 端口位 n 内部上拉器件关闭。 1 E 端口位 n 内部上拉器件使能。

	7	6	5	4	3	2	1	0
R W	PTESE7	PTESE6	PTESE5	PTESE4	PTESE3	PTESE2	PTESE1	PTESE0
复位	0	0	0	0	0	0	0	0

图 6-25. E 端口斜率控制使能寄存器 (PTESE)

表 6-24. PTESE 寄存器字段描述

字段	描述
7:0 PTESE[7:0]	**E 端口输出斜率使能位** — 这些控制位决定是否为相关 PTE 引脚使能输出斜率控制。对于配置为输入的 E 端口引脚，这些位不会产生任何影响。 0 E 端口位 n 输出斜率控制关闭。 1 E 端口位 n 输出斜率控制使能。

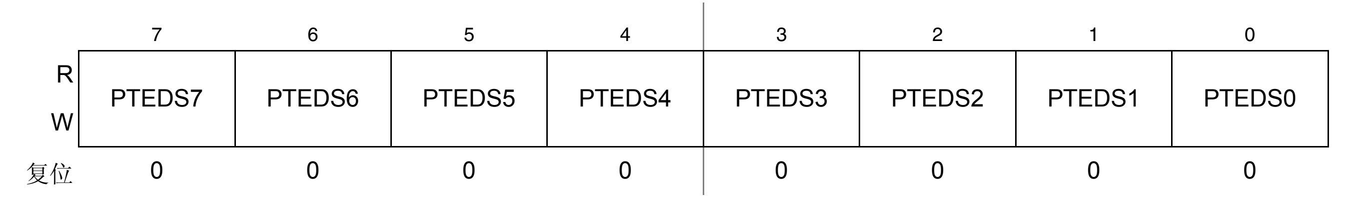


图 6-26. E 端口驱动强度选择寄存器 (PTEDS)

表 6-25. PTEDS 寄存器字段描述

字段	描述
7:0 PTEDS[7:0]	**E 端口输出驱动强度选择位** — 这些控制位为 PTE 引脚设置低 / 高输出驱动。 0 E 端口位 n 低输出驱动使能。 1 E 端口位 n 高输出驱动使能。

6.6.11 F 端口 I/O 寄存器 (PTFD 和 PTFDD)

F 端口并行 I/O 功能受下列寄存器控制。

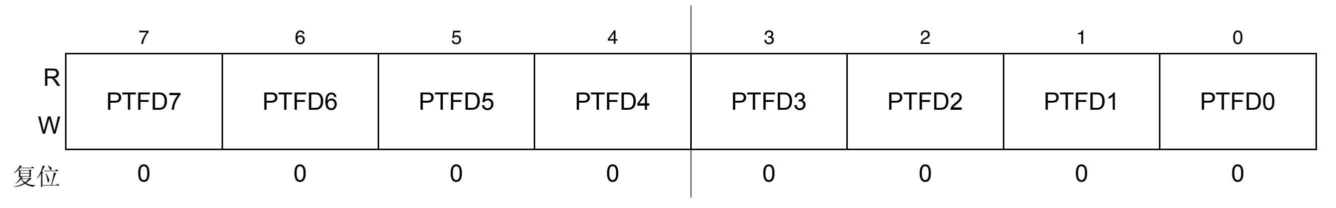


图 6-27. F 端口数据寄存器 (PTFD)

表 6-26. PTFD 寄存器字段描述

字段	描述
7:0 PTFDn	**F 端口数据寄存器位** — 对于配置为输入的 F 端口引脚，读数返回引脚上的逻辑电平。对于配置为输出的 F 端口引脚，读数返回最后一次写入寄存器的值。 写入值被锁存在本寄存器的所有位中。对于配置为输出的 F 端口引脚，逻辑电平驱动相应的 MCU 引脚。 复位强制 PTFD 都为 0，但是这些 0 未被驱出相应的引脚，因为复位也会将所有端口引脚配置为上拉关闭的高抗阻输入。

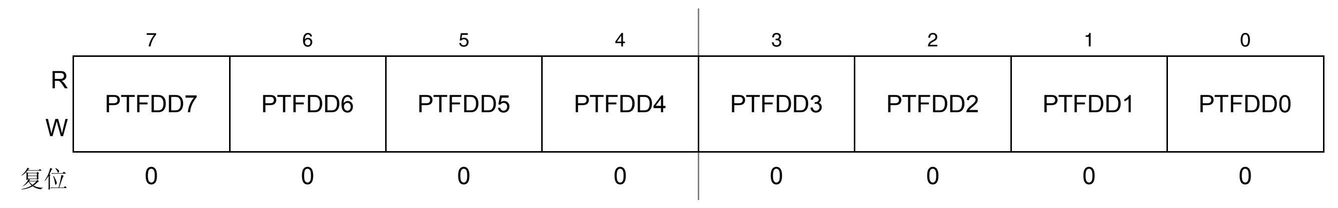


图 6-28. F 端口数据方向寄存器 (PTFDD)

表 6-27. PTFDD 寄存器字段描述

字段	描述
7:0 PTFDDn	**F 端口数据方向位** — 这些读 / 写位控制着 F 端口引脚的方向以及 PTFD 读取的内容。 0 输入 （输出驱动关闭），读数返回引脚值。 1 F 端口位 n 输出驱动使能，PTFD 读数返回 PTFDn 内容。

6.6.12 F 端口引脚控制寄存器 (PTFPE, PTFSE, PTFDS)

除了 I/O 控制外，F 端口引脚受下列的寄存器控制。

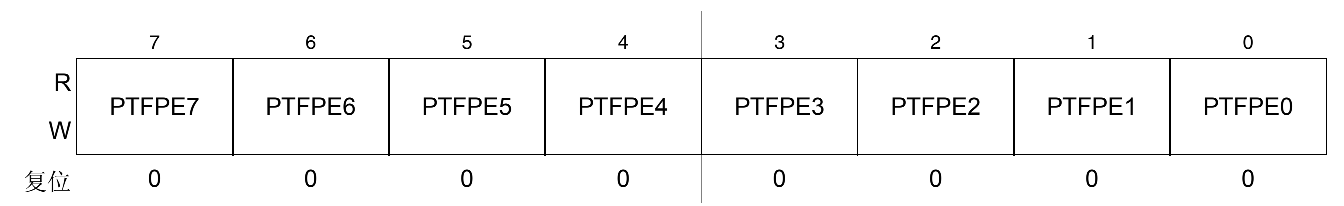


图 6-29. F 端口内部上拉使能寄存器 (PTFPE)

表 6-28. PTFPE 寄存器字段描述

字段	描述
7:0 PTFPEn	**F 端口内部上拉使能位** — 对于 PTF 引脚，这些控制位决定相关 PTF 引脚的内部上拉器件是否使能。对于配置为输出的 F 端口引脚，这些位不会产生影响，同时内部上拉器件关闭。 0 F 端口位 n 内部上拉器件关闭。 1 F 端口位 n 内部上拉器件使能。

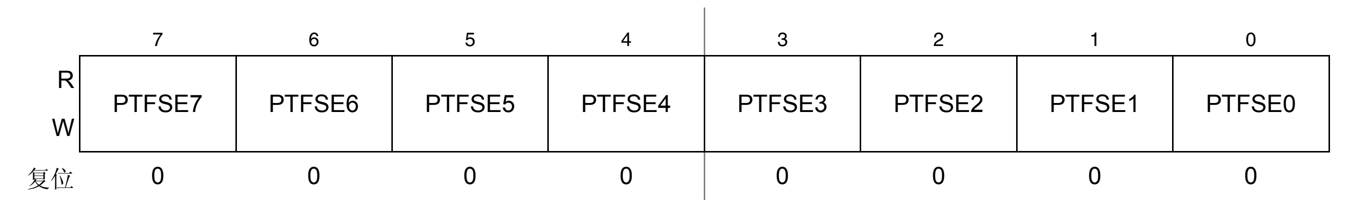


图 6-30. F 端口斜率控制使能寄存器 (PTFSE)

表 6-29. PTFSE 寄存器字段描述

字段	描述
7:0 PTFSEn	**F 端口输出斜率使能位** — 这些控制位决定是否为相关 PTF 引脚使能输出斜率控制。对于配置为输入的 F 端口引脚，这些位不会产生任何影响。 0 F 端口位 n 输出斜率控制关闭。 1 F 端口位 n 输出斜率控制使能。

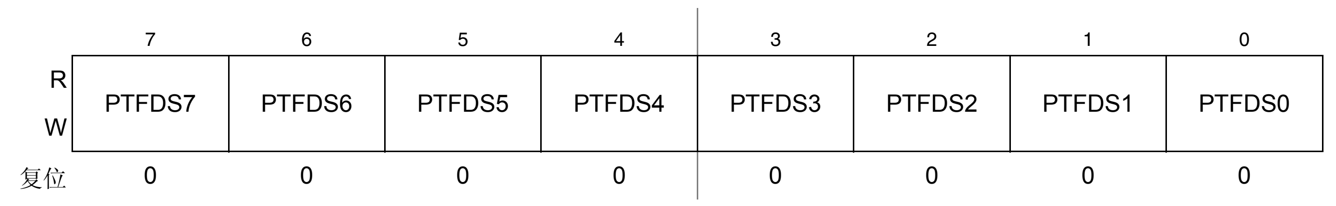


图 6-31. F 端口寄存器驱动强度选择 (PTFDS)

表 6-30. PTFDS 寄存器字段描述

字段	描述
7:0 PTFDSn	**F 端口输出驱动强度选择位** — 这些控制位为 PTF 引脚设置低 / 高输出驱动。 0 F 端口位 n 低输出驱动使能。 1 F 端口位 n 高输出驱动使能。

6.6.13 G 端口 I/O 寄存器 (PTGD 和 PTGDD)

G 端口并行 I/O 功能受下列寄存器控制。

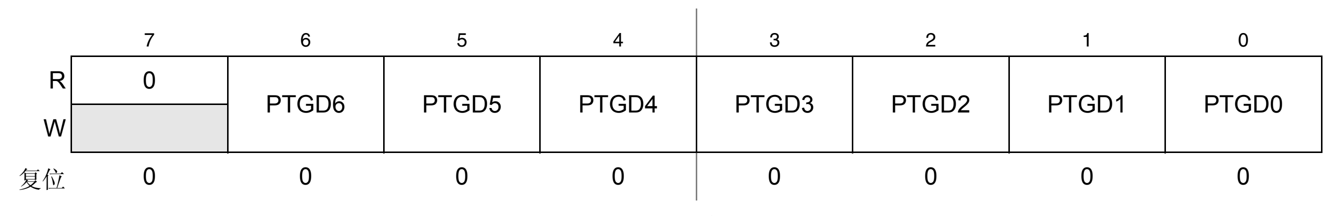


图 6-32. G 端口数据寄存器 (PTGD)

表 6-31. PTGD 寄存器字段描述

字段	描述
6:0 PTGD[6:0]	G 端口数据寄存器位 — 对于配置为输入的 G 端口引脚，读数返回引脚上的逻辑电平。对于配置为输出的 G 端口引脚，读数返回最后一次写入寄存器的值。 写入值被锁存在本寄存器的所有位中。对于配置为输出的 G 端口引脚，逻辑电平驱动相应的 MCU 引脚。 复位强制 PTGD 都为 0，但是这些 0 未被驱出相应的引脚，因为复位也会将所有端口引脚配置为上拉关闭的高抗阻输入。

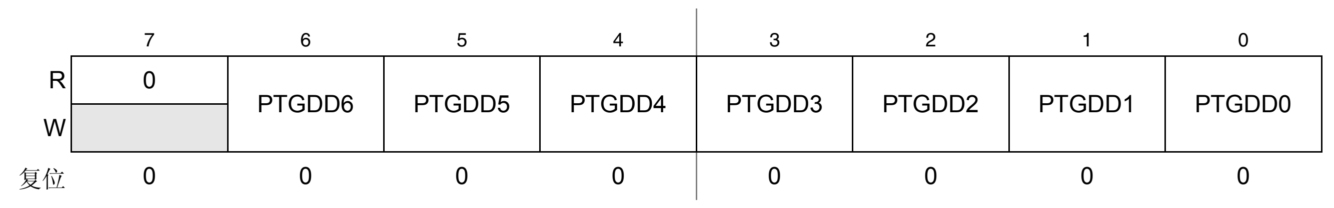


图 6-33. G 端口数据方向寄存器 (PTGDD)

表 6-32. PTGDD 寄存器字段描述

字段	描述
6:0 PTGDD[6:0]	**G 端口数据方向位** — 这些读 / 写位控制着 G 端口引脚的方向以及 PTGD 读取的内容。 0 输入 （输出驱动关闭），读数返回引脚值。 1 G 端口位 n 输出驱动使能，PTGD 读数返回 PTGDn 内容。

6.6.14 G 端口引脚 控制寄存器 (PTGPE, PTGSE, PTGDS)

除了 I/O 控制外，G 端口引脚还受下列寄存器控制。

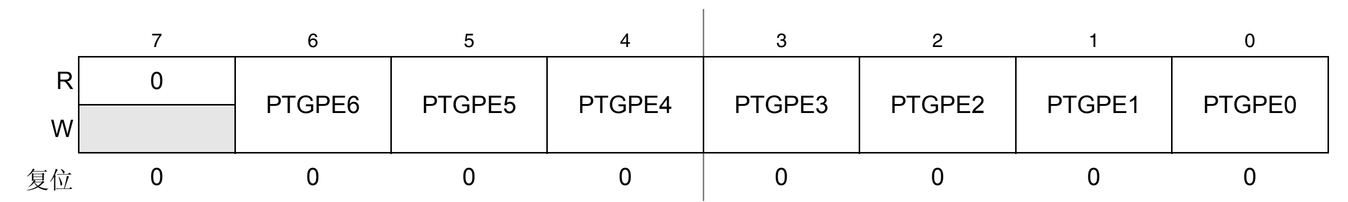


图 6-34. G 端口内部上拉使能寄存器 (PTGPE)

表 6-33. PTGPE 寄存器字段描述

字段	描述
6:0 PTGPE[6:0]	**G 端口内部上拉使能位** — 对于 PTG 引脚，这些控制位决定相关 PTG 引脚的内部上拉器件是否使能。对于配置为输出的 G 端口引脚，这些位不会产生影响，同时内部上拉器件关闭。 0 G 端口位 n 内部上拉器件关闭。 1 G 端口位 n 内部上拉器件使能。

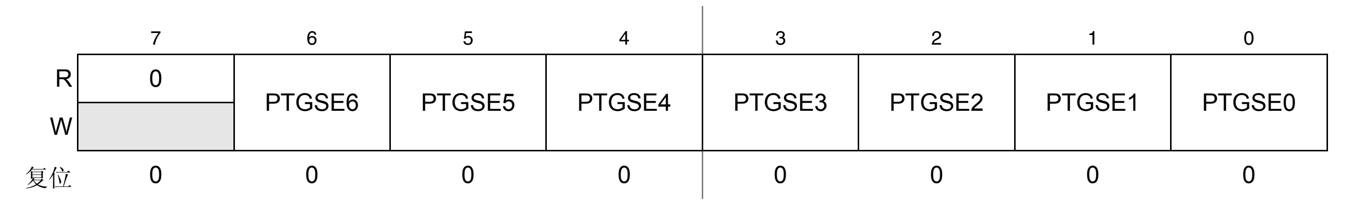


图 6-35. G 端口斜率控制使能寄存器 (PTGSE)

表 6-34. PTGSE 寄存器字段描述

字段	描述
6:0 PTGSE[6:0]	**G 端口输出斜率使能位** — 这些控制位决定是否为相关 PTG 引脚使能输出斜率控制。对于配置为输入的 G 端口引脚，这些位不会产生任何影响。 0 G 端口位 n 输出斜率控制关闭。 1 G 端口位 n 输出斜率控制使能。

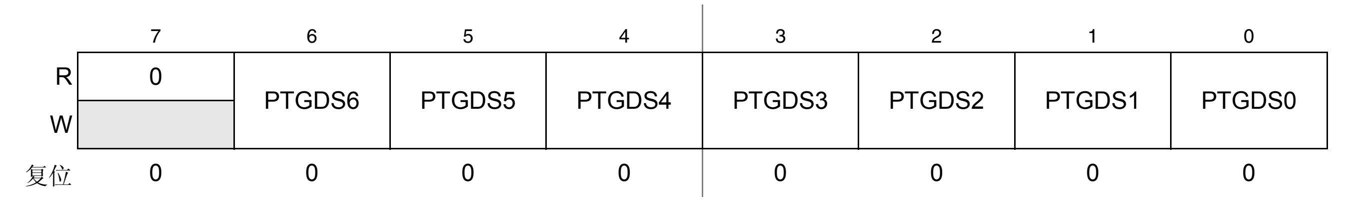


图 6-36. G 端口驱动强度选择寄存器 (PTGDS)

表 6-35. PTGDS 寄存器字段描述

字段	描述
6:0 PTGDS[6:0]	**G 端口输出驱动强度选择位** — 这些控制位为 PTG 引脚设置低 / 高输出驱动。 0 G 端口位 n 低输出驱动使能。 1 G 端口位 n 高输出驱动使能。

第 7 章
中央处理单元 （S08CPUV2）

7.1 概述

本节介绍了 HCS08 系列 CPU 的寄存器、寻址方式和指令集。更详细的信息请参考 《HCS08 Family Reference Manual》，该文件序号为 HCS08RMV1/D。

HCS08 CPU 与 M68HC08 CPU 的源和目标代码完全兼容。新增的指令和增强寻址模式提高了 C 编译器的效率，并且支持新的后台调试系统以代替早期 68HC08 的监控模式。

7.1.1 特性

HCS08 CPU 具有以下特性：

- 目标代码完全兼容 M68HC05 和 M68HC08 家族
- 所有寄存器和存储器映射到一个独立的 64 KB 的地址空间
- 16 位堆栈指针 （64 K 字节地址空间内任意大小、任意地址的堆栈）
- 16 位变址寄存器 （H:X）支持强大的索引地址模式
- 8 位累加器 （A）
- 许多指令把 X 作为第二个通用 8 位寄存器
- 7 种寻址模式：
  - 固有寻址模式 — 操作数存于内部寄存器
  - 相对寻址模式 — 8 位有符号偏移量的分支地址
  - 立即寻址模式 — 操作数位于下一个目标代码
  - 直接寻址模式 — 操作数位于 0x0000 到 0x00FF 之间
  - 扩展寻址模式 — 操作数位于 64K 字节地址空间内
  - H:X 相对变址寻址模式 — 提供包括自动增量在内的 5 种子模式
  - SP 相对变址寻址模式 — 大大提高 C 语言编译的效率
- 提供四种寻址模式组合的寄存器 - 寄存器数据转移指令
- 溢出、半进位、负、零和进位状况码支持根据带符号、无符号、 BCD 码操作的结果进行条件转移
- 高效率的位操作指令
- 快速 8 位乘 8 位和 16 位除 8 位指令
- STOP 和 WAIT 指令调用低功耗运行模式

7.2 编程模型和 CPU 寄存器

图 7-1 为 5 个 CPU 寄存器，微处理器的存储器不包括 CPU 寄存器。

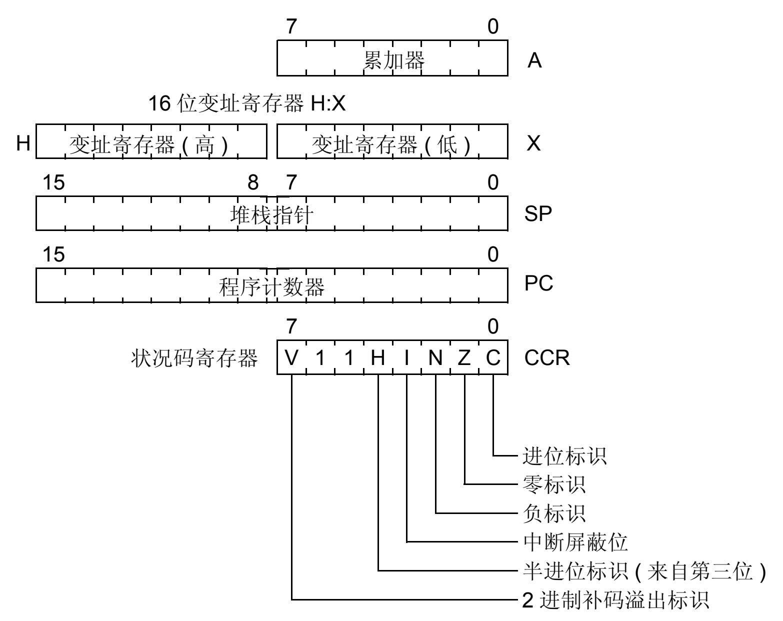


图 7-1. CPU 寄存器

7.2.1 累加器 (A)

这个累加器 (A) 是通用的 8 位寄存器。一个操作数经过累加器输入到算术逻辑单元 （ ALU ），在算术和逻辑运算后， ALU 的结果通常存放入 A 累加器。累加器可以用不同的寻址模式指定地址从存储器中装载数据，或者累加器 A 中的数据可以根据不同的寻址模式指定的地址装入到存储器中。

复位对累加器中的数据无影响。

7.2.2 变址寄存器 （H:X）

这个 16 位寄存器实际上是由两个 8 位寄存器 （H,X）组成，常一起作为一个 16 位地址指针。其中， H 装地址的高字节， X 装地址的低字节。所有变址寻址方式指令用 H:X 中的 16 位的值作为索引参考指针。但是，为了和早期 M68HC05 系列兼容，部分指令只能在低 8 位 （X）上运行。

许多指令把 X 作为第二个通用 8 位寄存器来存储 8 位数据。 X 可以被清零，增加，递减，取反，移位，循环移位。在算术逻辑操作时，转移指令用于 X 与累加器 A 进行数据交换。

为了和早期 M68HC05 系列兼容， H 在复位时被设为 0X00。复位对 X 无影响。

7.2.3 堆栈指针 （SP）

在一个自动后进先出 (LIFO) 堆栈中，该 16 位地址指针寄存器指向下一个可用地址。堆栈可以映射到有 RAM 的 64K 字节空间内的任意地址，它可以与所提供的 RAM 大小相同。堆栈可以自动保存子程序调用的返回地址，中断操作中的返回地址和 CPU 寄存器值，以及本地变量。 AIS （Add Immediate to Stack pointer）指令赋予一个 8 位有符号立即数给 SP。这种方法经常被用于为本地堆栈变量在堆栈中分配或回收空间。

为了兼容早期的 M68HC05 系列， SP 复位后被设为 0xFF。在复位初始化时，HCS08 程序通常将 SP 的值改为 RAM 最后地址值 （最高地址），来释放直接页上的 RAM(片上寄存器结束处一直到 0x00FF）。

RSP （复位堆栈指针）是与 M68HC05 系列兼容的指令，因为它仅影响堆栈指针的低位地址，所以很少用在 HCS08 程序中。

7.2.4 程序计数器 （PC）

程序计数器是一个 16 位寄存器，包含下一条指令或待取操作数的地址。

通常在程序执行期间，程序计数器在每次数据或指令被取出时会自动增加到存储器的下个连续位置。跳转、转移、中断和返回操作向程序计数器装载一个地址而不是下个连续位置。这就是流程的改变。

复位时，程序计数器从 0XFFFE 和 0XFFF 处装载复位向量。向量存储位置的第一个指令将在退出复位状态后被执行。

7.2.5 条件码寄存器 (CCR)

8 位条件码寄存器包含中断屏蔽位 I 和 5 个标识最近执行指令结果的状态标志位。第 6 位和第 5 位恒为逻辑 1。下图简要描述了 CCR 每个位的信息及功能。关于如何使用指令设置 CCR 的每个位，请参阅 HCS08 系列参考手册，第 1 卷，其文档号为 HCS08RMv1。

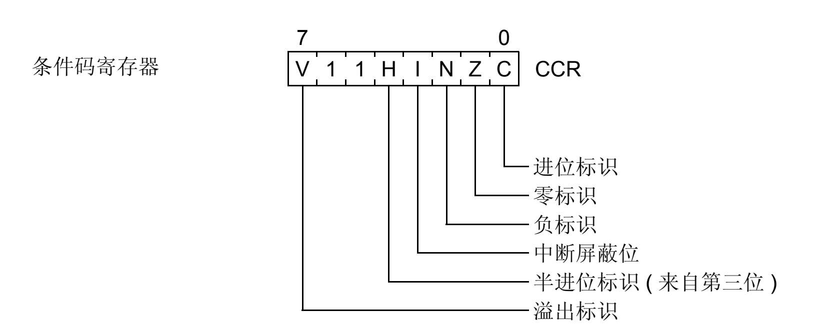


图 7-2. 条形码寄存器 （CCR）

表 7-1. CCR 寄存器位功能描述

位置	描述
7 V	二进制补码溢出标志位—当二进制补码溢出时， CPU 将设置溢出标志位。有符号指令 BGT、 BGE、 BLE、 BLT 使用溢出标志。 0 无溢出 1 溢出
4 H	半进位标志 — 执行加法指令 （ADD）和带进位加法指令 （ADC）时，如果累加器第 3 和第 4 位有进位， CPU 设置半进位标志。半进位标志在 BCD 码算术运算中很有用。 DAA 指令用 H 和 C 的状态来自动调整前面 ADD 或者 ADC 的结果，修正结果得到有效的 BCD 值。 0 第 3 位和第 4 位之间无进位 1 第 3 位和第 4 位之间进位
3 I	中断屏蔽位 — 当中断屏蔽位被置位时，禁止 CPU 中断。当中断屏蔽位被清除，开放 CPU 中断。当中断发生时，在第一个中断服务例程被执行之前， CPU 寄存器值被保存到堆栈后中断屏蔽被自动设置。在任何指令正在清 I （CLI、 TAP）时，中断不被认可。这确保了 CLI 或 TAP 的下一个指令执行时不被干扰。 0 允许中断 1 禁止中断
2 N	负标志—CPU 运算时，如果产生负结果则将负标志置 1，设置位 7 的结果。如果装载或存储值的较多重要位为 1 的话，简单的装载或存储 8 位或 16 位值也会引起 N 置 1 。

表 7-1. CCR 寄存器位功能描述 (续)

1 Z	零标志—CPU 进行运算过程中，如果数据或运算结果为零，零标志置 1，否者置零。 0 无零结果 1 有零结果
0 C	进位 / 借位标志—当进行加法时，在最高位 D7 上有进位；或在进行减法运算时需要向更高位借位，则 CPU 将进位 / 借位标志 C 置 1。一些指令如位测试，跳转，移位指令等也会影响该标志。 0 无进位 / 借位 1 有进位 / 借位

7.3 寻址模式

寻址模式决定 CPU 存取数据和操作数的方法。在 HCS08 系列芯片中，所有的存储器、状态和控制寄存器和 I/O 口共享一个 64K 字节线性地址空间，所以一个 16 位二进制地址可以唯一确定一个存储位置。这样，访问 RAM 变量的指令同时也能访问 I/O 和控制寄存器或非易失性程序空间。

某些指令的寻址方式不止一种。例如， MOVE 指令用一种寻址方式来指定源操作数，又用另一种寻址方式来确定目标地址。BRCLR、BRSET、CBEQ、DBNZ 等指令用一种寻址方式来确定一个操作数地址，并进行测试，如果假设条件为真时，就用相对寻址方式来确定分支的目标地址。对于 BRCLR、 BRSET、 CBEQ、DBNZ，列在指令设置表上的寻址方式用于定位需要测试的操作数，相对寻址方式用来定位分支目标地址。

7.3.1 固有寻址模式 （INH）

固有寻址模式所需的操作数已经存于 CPU 寄存器中，所以 CPU 不需要访问存储器来获取操作数。

7.3.2 相对寻址模式 （REL）

相对寻址模式用于确定分支指令的目的地址。 8 位有符号的偏移值在存储器中位于操作码之后。在执行指令时，如果分支条件为真，有符号偏移值扩展为 16 位有符号值且加到当前程序计数器，这样程序转移到分支目标地址处运行。

7.3.3 立即寻址模式 (IMM)

立即寻址模式中，用以完成指令的操作数包含在目标代码内，并紧跟着指令操作码存放在存储器中。在 16 位立即操作数情况下，高位字节存放在操作码之后的第一个存储位置，低位字节存放在下个存储位置。

7.3.4 直接寻址模式 (DIR)

直接寻址模式中，指令包含的地址低 8 位存放在直接页 （0x0000-0x00FF）。在执行指令时，通过连接 0x00 作为高位地址和指令的直接地址得到一个 16 位地址以获取目标操作数。这比给操作数分配一个完整的 16 位地址要更快，存取效率更高。

7.3.5 扩展寻址模式 (EXT)

在扩展寻址模式中，操作数的 16 位地址位于操作码后的两字节 （高位在先）。

7.3.6 变址寻址模式

变址寻址模式包含七种，其中五种使用 16 位 H:X 变址寄存器，另两种使用堆栈作为基值参考。

7.3.6.1 无偏移量变址模式 (IX)

在这种变址模式中，变址寄存器 H:X 的内容被用于访问操作数的地址。

7.3.6.2 无偏移量变址、变址自动加 1 寻址模式 (IX+)

在这种变址模式中， H:X 变址寄存器存储指令的操作数。当取出地址后， H:X 寄存器自动加 1(H:X=H:X+0x0001)。这种寻址模式只应用于 CBEQ 和 MOV 指令。

7.3.6.3 8 位偏移量变址模式 (IX1)

在这种变址模式中，一个无符号的 8 位偏移量与 H:X 寄存器相加，得到要访问的操作数地址。

7.3.6.4 8 位偏移量变址、变址自动加 1 寻址模式 (IX1+)

在这种变址模式中，一个无符号的 8 位偏移量与 H:X 寄存器相加，得到要访问的操作数地址。在操作数被取出后变址寄存器自动加 1。 CBEQ 是唯一使用这种寻址模式的指令。

7.3.6.5 16 位偏移量变址模式 (IX2)

在这种变址模式中，一个无符号的 16 位偏移量与 H:X 寄存器相加，得到要访问的操作数地址

7.3.6.6 8 位偏移量堆栈寻址模式 (SP1)

在这种变址模式中，指令集提供一个无符号的 8 位偏移量与堆栈指针 (SP) 中的 16 位值相加，得到要访问的操作数地址。

7.3.6.7 16 位偏移量堆栈寻址模式 (SP2)

在这种变址模式中，无符号的 16 位偏移量与堆栈指针 (SP) 中的 16 位值相加，得到要访问的操作数地址。

7.4 特殊操作

CPU 可以执行一些操作，这些操作与其他的 CPU 指令相似，只是没有操作码。另外一些指令如 STOP，WAIT 直接影响其他 MCU 的电路。本节详细将讲述这些操作。

7.4.1 复位序列

复位可以由上电复位、 COP 看门狗超时、外部复位引脚引起。当复位事件发生时，复位事件强制 CPU 立即停止正在执行的任何事件 (MCU 不会等待指令边界，再响应复位)。关于 MCU 如何确认复位及其来源的信息，可参阅复位、中断和系统配置章节。

当用以确定复位源是否来自内部的序列被执行，同时复位引脚不再有强制复位信号时，复位事件被认为结束。然后， CPU 执行一个 6 周期的序列从 0XFFFE 和 0XFFFF 处取出复位向量填充到指令队列，为执行程序作好准备。

7.4.2 中断序列

当中断请求发生时，在响应中断之前 CPU 先完成当前指令。这时程序计数器指向下条指令的开始位置，这也就是 CPU 中断返回地址。 CPU 对中断的响应是执行与软中断 （SWI）相同的操作序列，除非当中断序列开始时，由挂起的高优先级中断决定向量获取的地址。

CPU 中断序列如下：

1. 按顺序存储 PCL、 PCH、 X、 A、 CCR 的内容到堆栈中。
2. 设置 CCR 中的 I 位。
3. 获取中断向量高位。
4. 获取中断向量低位。
5. 延迟一个空闲总线周期。
6. 将 3 字节的程序信息 (其首地址由中断向量决定) 填入指令队列为执行中断服务例程第一条指令作准备。

CCR 中的内容被压入堆栈后，在进行中断服务例程时设置 CCR 的 I 位防止响应其它中断。虽然也可以通过指令在中断服务例程中清 I，但是有可能导致嵌套中断 （不推荐采用这种方法，因为这将增加程序调试和维护的难度）。

为了和早期的 M68HC05MCU 兼容， H:X 变址寄存器中的 H 不作为中断序列的一部分保存到堆栈中。用户必须在终端服务例程开始时用 PSHH 指令保存 H 内容，然后在 RTI 返回之前用 PULH 指令将 H 取出。如果能确定中断服务例程不使用任何与 H 相关的指令或自增寻址方式， H 值也可以不保存。

软中断除了不能被 CCR 中的 I 位屏蔽，其它和硬中断一样。并且，它和程序中的一个指令操作码相关联，所以它不是异步程序操作。

7.4.3 等待模式

WAIT 指令通过清零 CCR 的 I 位使能中断。然后关闭 CPU 时钟以节省功耗， CPU 维持在低功耗状态直到被中断或复位唤醒。当中断或复位发生时， CPU 从等待模式被唤醒。此时， CPU 重启内部时钟，并响应中断以及复位。

当 CPU 在等待模式时，如果一个串行的后台调试指令通过后台调试界面发送到 MCU， CPU 时钟将被重启并进入能处理其它串行背景指令的活跃背景模式。这样，即使在等待模式中主机开发系统仍能进入目标 MCU。

7.4.4 停止模式

通常，在停止模式下，系统的所有时钟包括晶振都被关闭以实现最小功耗。在这样的系统中，需要一个外部电路来计算时间，用于在指定的时间产生信号唤醒 MCU。和早期的 M68HC05 和 M68HC08 不同，HCS08 在停止模式中可以设定运行最少时钟。这就允许一个内部周期信号从停止模式中唤醒 MCU。

当一个主机调试系统连接到后台调试引脚 （BKGD）且通过后台调试接口串行命令设置 ENBDM 位时（或者 MCU 在复位时进入活跃的后台调试模式）， MCU 进入停止模式时振荡器被强制活跃。在这种情况下，当 CPU 在停止模式下通过后台调试接口向 MCU 发送后台调试指令， CPU 时钟将被开启并进入能处理其它串行背景指令的活跃背景模式。这样，即使在停止模式中主机开发系统仍能进入目标 MCU。

从停止模式恢复的方式取决于具体的 HCS08 型号以及振荡器是否在停止模式时停止。详细信息请参阅操作模式章节。

7.4.5 背景模式

与 M68HC08 相比， HCS08 新增了 BGND 指令。 BGND 通常用在普通用户程序中，强迫 CPU 停止处理当前指令进入活跃的背景模式。重新继续运行用户程序的唯一方式是通过复位或主机调试系统的后台调试接口发出 GO、 TRACE1 或者 TAGGO 串行命令。

可以用后台调试操作码替换目标断点地址操作码来设置软断点。当程序到达断点地址时， CPU 强制进入活跃背景模式而不是继续用户程序。

7.5 HCS08 指令设置摘要

表 7-2 包含了 HCS08 所有寻址模式的指令集。本表提供了各操作数的结构、在内部总线周期的执行时间、以及每个指令在每种寻址模式的循环方式。

表 7-2. 指令集摘要 (第 1 页，共 9 页)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	I N Z C
ADC *#opr8i* ADC *opr8a* ADC *opr16a* ADC *oprx16*,X ADC *oprx8*,X ADC ,X ADC *oprx16*,SP ADC *oprx8*,SP	进位加 A ← (A) + (M) + (C)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A9 ii B9 dd C9 hh ll D9 ee ff E9 ff F9 9E D9 ee ff 9E E9 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	↕ 1 1 ↕	– ↕ ↕ ↕
ADD *#opr8i* ADD *opr8a* ADD *opr16a* ADD *oprx16*,X ADD *oprx8*,X ADD ,X ADD *oprx16*,SP ADD *oprx8*,SP	无进位加 A ← (A) + (M)	IMM DIR EXT IX2 IX1 IX SP2 SP1	AB ii BB dd CB hh ll DB ee ff EB ff FB 9E DB ee ff 9E EB ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	↕ 1 1 ↕	– ↕ ↕ ↕
AIS *#opr8i*	将立即数 （有符号）压入堆栈 SP ← (SP) + (M)	IMM	A7 ii	2	pp	– 1 1 –	– – – –
AIX *#opr8i*	将立即数 （有符号）压入 编制寄存器 （H:X） H:X ← (H:X) + (M)	IMM	AF ii	2	pp	– 1 1 –	– – – –
AND *#opr8i* AND *opr8a* AND *opr16a* AND *oprx16*,X AND *oprx8*,X AND ,X AND *oprx16*,SP AND *oprx8*,SP	逻辑与 A ← (A) & (M)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A4 ii B4 dd C4 hh ll D4 ee ff E4 ff F4 9E D4 ee ff 9E E4 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 –	– ↕ ↕ –
ASL *opr8a* ASLA ASLX ASL *oprx8*,X ASL ,X ASL *oprx8*,SP	算术左移 C ← b7 … b0 ← 0 (和 LSL 相同)	DIR INH INH IX1 IX SP1	38 dd 48 58 68 ff 78 9E 68 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ 1 1 –	– ↕ ↕ ↕
ASR *opr8a* ASRA ASRX ASR *oprx8*,X ASR ,X ASR *oprx8*,SP	算术右移 b7 … b0 → C	DIR INH INH IX1 IX SP1	37 dd 47 57 67 ff 77 9E 67 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ 1 1 –	– ↕ ↕ ↕
BCC *rel*	进位位清零则转移 (如果 C = 0)	REL	24 rr	3	ppp	– 1 1 –	– – – –

表 7-2. 指令集摘要 (第 2 页，共 9 页)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	I N Z C
BCLR *n*,*opr8a*	存储器中位 n 清零 (Mn ← 0)	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 dd 13 dd 15 dd 17 dd 19 dd 1B dd 1D dd 1F dd	5 5 5 5 5 5 5 5	rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp	– 1 1 –	– – – –
BCS *rel*	进位位置 1 则转移 (如果 C = 1) (和 BLO 相同)	REL	25 rr	3	ppp	– 1 1 –	– – – –
BEQ *rel*	相等则转移 (如果 Z = 1)	REL	27 rr	3	ppp	– 1 1 –	– – – –
BGE *rel*	大于等于则转移 (if N ⊕ V = 0) (带符号)	REL	90 rr	3	ppp	– 1 1 –	– – – –
BGND	ENBDM = 1 则进入活跃后台调试状态 等待和处理 BDM 指令直到 GO, TRACE1, TAGGO	INH	82	5+	fp...ppp	– 1 1 –	– – – –
BGT *rel*	大于则转移 (如果 Z | (N ⊕ V) = 0) (带符号)	REL	92 rr	3	ppp	– 1 1 –	– – – –
BHCC *rel*	半进位清零则转移 (如果 H = 0)	REL	28 rr	3	ppp	– 1 1 –	– – – –
BHCS *rel*	半进位置 1 则转移 (如果 H = 1)	REL	29 rr	3	ppp	– 1 1 –	– – – –
BHI *rel*	为高则转移 (如果 C | Z = 0)	REL	22 rr	3	ppp	– 1 1 –	– – – –
BHS *rel*	高于或相同则转移 (如果 C = 0) (和 BCC 相同)	REL	24 rr	3	ppp	– 1 1 –	– – – –
BIH *rel*	IRQ 引脚为高则转移 (如果 IRQ pin = 1)	REL	2F rr	3	ppp	– 1 1 –	– – – –
BIL *rel*	IRQ 引脚为低则转移 (如果 IRQ pin = 0)	REL	2E rr	3	ppp	– 1 1 –	– – – –
BIT #*opr8i* BIT *opr8a* BIT *opr16a* BIT *oprx16*,X BIT *oprx8*,X BIT ,X BIT *oprx16*,SP BIT *oprx8*,SP	位测试 (A) & (M) (CCR 更新但操作数不变)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A5 ii B5 dd C5 hh ll D5 ee ff E5 ff F5 9E D5 ee ff 9E E5 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 –	– ↕ ↕ –
BLE *rel*	小于等于则转移 (如果 Z | (N ⊕ V) = 1) (带符号)	REL	93 rr	3	ppp	– 1 1 –	– – – –
BLO *rel*	为低则转移 (如果 C = 1) (和 BCS 相同)	REL	25 rr	3	ppp	– 1 1 –	– – – –
BLS *rel*	为低或相同则转移 (如果 C | Z = 1)	REL	23 rr	3	ppp	– 1 1 –	– – – –
BLT *rel*	小于则转移 (如果 N ⊕ V = 1) (带符号)	REL	91 rr	3	ppp	– 1 1 –	– – – –
BMC *rel*	中断屏蔽位清零则转移 (如果 I = 0)	REL	2C rr	3	ppp	– 1 1 –	– – – –
BMI *rel*	负则转移 (如果 N = 1)	REL	2B rr	3	ppp	– 1 1 –	– – – –
BMS *rel*	中断屏蔽位置 1 则转移 (如果 I = 1)	REL	2D rr	3	ppp	– 1 1 –	– – – –
BNE *rel*	不等则转移 (如果 Z = 0)	REL	26 rr	3	ppp	– 1 1 –	– – – –
BPL *rel*	正则转移 (如果 N = 0)	REL	2A rr	3	ppp	– 1 1 –	– – – –
BRA *rel*	一直转移 (如果 I = 1)	REL	20 rr	3	ppp	– 1 1 –	– – – –

表 7-2. 指令集摘要 (第 3 页，共 9 页)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	I N Z C
BRCLR *n*,*opr8a*,*rel*	存储器中位 n 清零 (如果 (Mn) = 0)	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 dd rr 03 dd rr 05 dd rr 07 dd rr 09 dd rr 0B dd rr 0D dd rr 0F dd rr	5 5 5 5 5 5 5 5	rpppp rpppp rpppp rpppp rpppp rpppp rpppp rpppp	– 1 1 –	– – – ↕
BRN *rel*	从不转移 (如果 I = 0)	REL	21 rr	3	ppp	– 1 1 –	– – – –
BRSET *n*,*opr8a*,*rel*	存储器中位 n 置 1 则转移 (如果 (Mn) = 1)	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 dd rr 02 dd rr 04 dd rr 06 dd rr 08 dd rr 0A dd rr 0C dd rr 0E dd rr	5 5 5 5 5 5 5 5	rpppp rpppp rpppp rpppp rpppp rpppp rpppp rpppp	– 1 1 –	– – – ↕
BSET *n*,*opr8a*	存储器中位 n 置 1 (Mn ← 1)	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 dd 12 dd 14 dd 16 dd 18 dd 1A dd 1C dd 1E dd	5 5 5 5 5 5 5 5	rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp rfwpp	– 1 1 –	– – – –
BSR *rel*	转移到子程序 PC ← (PC) + $0002 push (PCL); SP ← (SP) – $0001 push (PCH); SP ← (SP) – $0001 PC ← (PC) + *rel*	REL	AD rr	5	ssppp	– 1 1 –	– – – –
CBEQ *opr8a*,*rel* CBEQA #*opr8i*,*rel* CBEQX #*opr8i*,*rel* CBEQ *oprx8*,X+,*rel* CBEQ ,X+,*rel* CBEQ *oprx8*,SP,*rel*	相等则比较转移 如果 (A) = (M) 则转移 如果 (A) = (M) 则转移 如果 (X) = (M) 则转移 如果 (A) = (M) 则转移 如果 (A) = (M) 则转移 如果 (A) = (M) 则转移	DIR IMM IMM IX1+ IX+ SP1	31 dd rr 41 ii rr 51 ii rr 61 ff rr 71 rr 9E 61 ff rr	5 4 4 5 5 6	rpppp pppp pppp rpppp rfppp prpppp	– 1 1 –	– – – –
CLC	清进位位 (C ← 0)	INH	98	1	p	– 1 1 –	– – – 0
CLI	清中断屏蔽位 (I ← 0)	INH	9A	1	p	– 1 1 –	0 – – –
CLR *opr8a* CLRA CLRX CLRH CLR *oprx8*,X CLR ,X CLR *oprx8*,SP	清位 M ← $00 A ← $00 X ← $00 H ← $00 M ← $00 M ← $00 M ← $00	DIR INH INH INH IX1 IX SP1	3F dd 4F 5F 8C 6F ff 7F 9E 6F ff	5 1 1 1 5 4 6	rfwpp p p p rfwpp rfwp prfwpp	0 1 1 –	– 0 1 –

表 7-2. 指令集摘要 (第 4 页，共 9 页)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	I N Z C
CMP #*opr8i* CMP *opr8a* CMP *opr16a* CMP *oprx16*,X CMP *oprx8*,X CMP ,X CMP *oprx16*,SP CMP *oprx8*,SP	存储器与累加器相比 A – M (CCR 更新但操作数不变)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A1 ii B1 dd C1 hh ll D1 ee ff E1 ff F1 9E D1 ee ff 9E E1 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	↕ 1 1 –	– ↕ ↕ ↕
COM *opr8a* COMA COMX COM *oprx8*,X COM ,X COM *oprx8*,SP	求补 M ← ($\overline{M}$)= $FF – (M) (一个数的补)A ← ($\overline{A}$) = $FF – (A) X ← ($\overline{X}$) = $FF – (X) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M)	DIR INH INH IX1 IX SP1	33 dd 43 53 63 ff 73 9E 63 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	0 1 1 –	– ↕ ↕ 1
CPHX *opr16a* CPHX #*opr16i* CPHX *opr8a* CPHX *oprx8*,SP	比较变址寄存器 (H:X) 和存储器 (H:X) – (M:M + $0001) (CCR 更新但操作数不变)	EXT IMM DIR SP1	3E hh ll 65 jj kk 75 dd 9E F3 ff	6 3 5 6	prrfpp ppp rrfpp prrfpp	↕ 1 1 –	– ↕ ↕ ↕
CPX #*opr8i* CPX *opr8a* CPX *opr16a* CPX *oprx16*,X CPX *oprx8*,X CPX ,X CPX *oprx16*,SP CPX *oprx8*,SP	比较 X (低位变址寄存器) 和存储器 X – M (CCR 更新但操作数不变)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A3 ii B3 dd C3 hh ll D3 ee ff E3 ff F3 9E D3 ee ff 9E E3 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	↕ 1 1 –	– ↕ ↕ ↕
DAA	在 BCD 码 ADD,ADC 操作后转换累加器内容到十进制	INH	72	1	p	U 1 1 –	– ↕ ↕ ↕
DBNZ *opr8a*,*rel* DBNZA *rel* DBNZX *rel* DBNZ *oprx8*,X,*rel* DBNZ ,X,*rel* DBNZ *oprx8*,SP,*rel*	如果不为零，则 A, X, 或者 M 自减并且转移 (如果 (result) ≠ 0) DBNZX 影响 X 但不影响 H	DIR INH INH IX1 IX SP1	3B dd rr 4B rr 5B rr 6B ff rr 7B rr 9E 6B ff rr	7 4 4 7 6 8	rfwpppp fppp fppp rfwpppp rfwppp prfwpppp	– 1 1 –	– – – –
DEC *opr8a* DECA DECX DEC *oprx8*,X DEC ,X DEC *oprx8*,SP	自减 M ← (M) – $01 A ← (A) – $01 X ← (X) – $01 M ← (M) – $01 M ← (M) – $01 M ← (M) – $01	DIR INH INH IX1 IX SP1	3A dd 4A 5A 6A ff 7A 9E 6A ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ 1 1 –	– ↕ ↕ –
DIV	除 A ← (H:A)÷(X); H ← 余数	INH	52	6	fffffp	– 1 1 –	– – ↕ ↕
EOR #*opr8i* EOR *opr8a* EOR *opr16a* EOR *oprx16*,X EOR *oprx8*,X EOR ,X EOR *oprx16*,SP EOR *oprx8*,SP	带累加器的存储器异或 A ← (A ⊕ M)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A8 ii B8 dd C8 hh ll D8 ee ff E8 ff F8 9E D8 ee ff 9E E8 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 –	– ↕ ↕ –

表 7-2. 指令集摘要 (第 5 页，共 9 页)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	I N Z C
INC *opr8a* INCA INCX INC *oprx8*,X INC ,X INC *oprx8*,SP	自增 M ← (M) + $01 A ← (A) + $01 X ← (X) + $01 M ← (M) + $01 M ← (M) + $01 M ← (M) + $01	DIR INH INH IX1 IX SP1	3C dd 4C 5C 6C ff 7C 9E 6C ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ 1 1 –	– ↕ ↕ –
JMP *opr8a* JMP *opr16a* JMP *oprx16*,X JMP *oprx8*,X JMP ,X	跳转 PC ← 跳转地址	DIR EXT IX2 IX1 IX	BC dd CC hh ll DC ee ff EC ff FC	3 4 4 3 3	ppp pppp pppp ppp ppp	– 1 1 –	– – – –
JSR *opr8a* JSR *opr16a* JSR *oprx16*,X JSR *oprx8*,X JSR ,X	跳转到子程序 PC ← (PC) + *n* (*n* = 1, 2, or 3) Push (PCL); SP ← (SP) – $0001 Push (PCH); SP ← (SP) – $0001 PC ← Unconditional Address	DIR EXT IX2 IX1 IX	BD dd CD hh ll DD ee ff ED ff FD	5 6 6 5 5	ssppp pssppp pssppp ssppp ssppp	– 1 1 –	– – – –
LDA *#opr8i* LDA *opr8a* LDA *opr16a* LDA *oprx16*,X LDA *oprx8*,X LDA ,X LDA *oprx16*,SP LDA *oprx8*,SP	从存储器装入累加器 A ← (M)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A6 ii B6 dd C6 hh ll D6 ee ff E6 ff F6 9E D6 ee ff 9E E6 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 –	– ↕ ↕ –
LDHX *#opr16i* LDHX *opr8a* LDHX *opr16a* LDHX ,X LDHX *oprx16*,X LDHX *oprx8*,X LDHX *oprx8*,SP	装入变址寄存器 （H:X） H:X ← (M:M + $0001)	IMM DIR EXT IX IX2 IX1 SP1	45 jj kk 55 dd 32 hh ll 9E AE 9E BE ee ff 9E CE ff 9E FE ff	3 4 5 5 6 5 5	ppp rrpp prrpp prrfp pprrpp prrpp prrpp	0 1 1 –	– ↕ ↕ –
LDX *#opr8i* LDX *opr8a* LDX *opr16a* LDX *oprx16*,X LDX *oprx8*,X LDX ,X LDX *oprx16*,SP LDX *oprx8*,SP	从存储器中装入 X （低位变址寄存器） X ← (M)	IMM DIR EXT IX2 IX1 IX SP2 SP1	AE ii BE dd CE hh ll DE ee ff EE ff FE 9E DE ee ff 9E EE ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 –	– ↕ ↕ –
LSL *opr8a* LSLA LSLX LSL *oprx8*,X LSL ,X LSL *oprx8*,SP	逻辑左移 C ← [b7 … b0] ← 0 (和 ASL 相同)	DIR INH INH IX1 IX SP1	38 dd 48 58 68 ff 78 9E 68 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ 1 1 –	– ↕ ↕ ↕
LSR *opr8a* LSRA LSR*X* LSR *oprx8*,X LSR ,X LSR *oprx8*,SP	逻辑右移 0 → [b7 … b0] → C	DIR INH INH IX1 IX SP1	34 dd 44 54 64 ff 74 9E 64 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ 1 1 –	– 0 ↕ ↕

表 7-2. 指令集摘要 (第 6 页，共 9 页)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR V 1 1 H	 I N Z C
MOV *opr8a*,*opr8a* MOV *opr8a*,X+ MOV #*opr8i*,*opr8a* MOV ,X+,*opr8a*	移动 $(M)_{destination} \leftarrow (M)_{source}$ In IX+/DIR and DIR/IX+ Modes, H:X ← (H:X) + $0001	DIR/DIR DIR/IX+ IMM/DIR IX+/DIR	4E dd dd 5E dd 6E ii dd 7E dd	5 5 4 5	rpwpp rfwpp pwpp rfwpp	0 1 1 –	– ↕ ↕ –
MUL	无符号相乘 X:A ← (X) × (A)	INH	42	5	ffffp	– 1 1 0	– – – 0
NEG *opr8a* NEGA NEGX NEG *oprx8*,X NEG ,X NEG *oprx8*,SP	取负 M ← – (M) = $00 – (M) (二进制补码)A ← – (A) = $00 – (A) X ← – (X) = $00 – (X) M ← – (M) = $00 – (M) M ← – (M) = $00 – (M) M ← – (M) = $00 – (M)	DIR INH INH IX1 IX SP1	30 dd 40 50 60 ff 70 9E 60 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ 1 1 –	– ↕ ↕ ↕
NOP	空操作 — 使用一个总线周期	INH	9D	1	p	– 1 1 –	– – – –
NSA	累加器半位元组交换 A ← (A[3:0]:A[7:4])	INH	62	1	p	– 1 1 –	– – – –
ORA #*opr8i* ORA *opr8a* ORA *opr16a* ORA *oprx16*,X ORA *oprx8*,X ORA ,X ORA *oprx16*,SP ORA *oprx8*,SP	累加器或存储器 A ← (A) | (M)	IMM DIR EXT IX2 IX1 IX SP2 SP1	AA ii BA dd CA hh ll DA ee ff EA ff FA 9E DA ee ff 9E EA ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	0 1 1 –	– ↕ ↕ –
PSHA	把累加器压入堆栈 Push (A); SP ← (SP) – $0001	INH	87	2	sp	– 1 1 –	– – – –
PSHH	把 H （高位变址寄存器）压入堆栈 Push (H); SP ← (SP) – $0001	INH	8B	2	sp	– 1 1 –	– – – –
PSHX	把 X （低位变址寄存器）压入堆栈 Push (X); SP ← (SP) – $0001	INH	89	2	sp	– 1 1 –	– – – –
PULA	累加器出栈 SP ← (SP + $0001); Pull (A)	INH	86	3	ufp	– 1 1 –	– – – –
PULH	H （高位变址寄存器）出栈 SP ← (SP + $0001); Pull (H)	INH	8A	3	ufp	– 1 1 –	– – – –
PULX	X （低位变址寄存器）出栈 SP ← (SP + $0001); Pull (X)	INH	88	3	ufp	– 1 1 –	– – – –
ROL *opr8a* ROLA ROLX ROL *oprx8*,X ROL ,X ROL *oprx8*,SP	进位循环左移 C ← b7 … b0 b7 b0	DIR INH INH IX1 IX SP1	39 dd 49 59 69 ff 79 9E 69 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ 1 1 –	– ↕ ↕ ↕
ROR *opr8a* RORA RORX ROR *oprx8*,X ROR ,X ROR *oprx8*,SP	进位循环右移 b7 … b0 → C b7 b0	DIR INH INH IX1 IX SP1	36 dd 46 56 66 ff 76 9E 66 ff	5 1 1 5 4 6	rfwpp p p rfwpp rfwp prfwpp	↕ 1 1 –	– ↕ ↕ ↕

表 7-2. 指令集摘要 (第 7 页，共 9 页)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	I N Z C
RSP	堆栈复位 (低位) SPL ← $FF (不影响高字节)	INH	9C	1	p	– 1 1 –	– – – –
RTI	中断返回 SP ← (SP) + $0001; Pull (CCR) SP ← (SP) + $0001; Pull (A) SP ← (SP) + $0001; Pull (X) SP ← (SP) + $0001; Pull (PCH) SP ← (SP) + $0001; Pull (PCL)	INH	80	9	uuuuufppp	↕ 1 1 ↕	↕ ↕ ↕ ↕
RTS	从子程序返回 SP ← SP + $0001; Pull (PCH) SP ← SP + $0001; Pull (PCL)	INH	81	5	ufppp	– 1 1 –	– – – –
SBC *#opr8i* SBC *opr8a* SBC *opr16a* SBC *oprx16*,X SBC *oprx8*,X SBC ,X SBC *oprx16*,SP SBC *oprx8*,SP	带进位减 A ← (A) – (M) – (C)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A2 ii B2 dd C2 hh ll D2 ee ff E2 ff F2 9E D2 ee ff 9E E2 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	↕ 1 1 –	– ↕ ↕ ↕
SEC	进位位置位 (C ← 1)	INH	99	1	p	– 1 1 –	– – – 1
SEI	中断屏蔽位置位 (I ← 1)	INH	9B	1	p	– 1 1 –	1 – – –
STA *opr8a* STA *opr16a* STA *oprx16*,X STA *oprx8*,X STA ,X STA *oprx16*,SP STA *oprx8*,SP	将累加器中内容存储到存储器 M ← (A)	DIR EXT IX2 IX1 IX SP2 SP1	B7 dd C7 hh ll D7 ee ff E7 ff F7 9E D7 ee ff 9E E7 ff	3 4 4 3 2 5 4	wpp pwpp pwpp wpp wp ppwpp pwpp	0 1 1 –	– ↕ ↕ –
STHX *opr8a* STHX *opr16a* STHX *oprx8*,SP	存储 H:X (变址寄存器) (M:M + $0001) ← (H:X)	DIR EXT SP1	35 dd 96 hh ll 9E FF ff	4 5 5	wwpp pwwpp pwwpp	0 1 1 –	– ↕ ↕ –
STOP	中断使能 停止处理 参考 MCU 文档 I bit ← 0; Stop Processing	INH	8E	2	fp...	– 1 1 –	0 – – –
STX *opr8a* STX *opr16a* STX *oprx16*,X STX *oprx8*,X STX ,X STX *oprx16*,SP STX *oprx8*,SP	存储 X （变址寄存器低 8 位）到存储器 M ← (X)	DIR EXT IX2 IX1 IX SP2 SP1	BF dd CF hh ll DF ee ff EF ff FF 9E DF ee ff 9E EF ff	3 4 4 3 2 5 4	wpp pwpp pwpp wpp wp ppwpp pwpp	0 1 1 –	– ↕ ↕ –

表 7-2. 指令集摘要 (第 8 页，共 9 页)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR	
						V 1 1 H	**I N Z C**
SUB #*opr8i* SUB *opr8a* SUB *opr16a* SUB *oprx16*,X SUB *oprx8*,X SUB ,X SUB *oprx16*,SP SUB *oprx8*,SP	减 A ← (A) – (M)	IMM DIR EXT IX2 IX1 IX SP2 SP1	A0 ii B0 dd C0 hh ll D0 ee ff E0 ff F0 9E D0 ee ff 9E E0 ff	2 3 4 4 3 3 5 4	pp rpp prpp prpp rpp rfp pprpp prpp	↕ 1 1 –	– ↕ ↕ ↕
SWI	软中断 PC ← (PC) + $0001 Push (PCL); SP ← (SP) – $0001 Push (PCH); SP ← (SP) – $0001 Push (X); SP ← (SP) – $0001 Push (A); SP ← (SP) – $0001 Push (CCR); SP ← (SP) – $0001 I ← 1; PCH ← 高字节中断矢量 PCL ← 低字节中断矢量	INH	83	11	sssssvvfppp	– 1 1 –	1 – – –
TAP	转移累加器到 CCR CCR ← (A)	INH	84	1	p	↕ 1 1 ↕	↕ ↕ ↕ ↕
TAX	转移累加器到 X (低位变址寄存器) X ← (A)	INH	97	1	p	– 1 1 –	– – – –
TPA	转移 CCR 到累加器 A ← (CCR)	INH	85	1	p	– 1 1 –	– – – –
TST *opr8a* TSTA TSTX TST *oprx8*,X TST ,X TST *oprx8*,SP	测试零或负数 (M) – $00 (A) – $00 (X) – $00 (M) – $00 (M) – $00 (M) – $00	DIR INH INH IX1 IX SP1	3D dd 4D 5D 6D ff 7D 9E 6D ff	4 1 1 4 3 5	rfpp p p rfpp rfp prfpp	0 1 1 –	– ↕ ↕ –
TSX	转移 SP 到变址寄存器 . H:X ← (SP) + $0001	INH	95	2	fp	– 1 1 –	– – – –
TXA	转移 X （低位变址寄存器）到累加器 A ← (X)	INH	9F	1	p	– 1 1 –	– – – –

表 7-2. 指令集摘要 (第 9 页，共 9 页)

源格式	操作	寻址方式	操作码	总线周期	循环细节	CCR V 1 1 H	CCR I N Z C
TXS	转移变址寄存器到 SP SP ← (H:X) – $0001	INH	94	2	fp	– 1 1 –	– – – –
WAIT	中断使能，等待中断 I bit ← 0; Halt CPU	INH	8F	2+	fp...	– 1 1 –	0 – – –

源格式栏中，除了用斜体表示的表达式外，其它信息必须以相同的格式出现在汇编源文件中。原始的 3 至 5 个助记符是一种文字表达。所有的逗号，井号 （#），圆括号，加号 （+）都是文字字符。
n — 任何估算 0-7 之间的单整形数的标识或表达式
opr8i — 任何估算 8 位立即数值的标识或表达式
opr16i — 任何估算 16 位立即数值的标识或表达式
opr8a — 任何估算 8 位直接页地址 （0x00XX）的标识或表达式。
opr16a — 任何估算 16 位地址的标识或表达式。
oprx8 — 任何估算无符号 8 位立即数值的标识或表达式 , 用于变址寻址
oprx16 — 任何用来估值 16 位立即数值的标识或表达式，用于变址寻址
rel — 任何标识或表达式所涉及地址在当前指令末子节目标操作码后的下个地址 起 -128 至 127 之间。

操作符 :

A	累加器
CCR	条件码寄存器
H	变址寄存器高 8 位
M	寄存器地址
n	任何位
opr	操作数 (1 或 2 字节)
PC	程序计数器
PCH	程序计数器高 8 位
PCL	程序计数器低 8 位
rel	相对程序计数器偏移字节
SP	堆栈指针
SPL	堆栈指针低 8 位
X	变址寄存器低 8 位
&	布尔与
|	布尔或
⊕	布尔异或
( )	寄存器或存储器内容显示在圆括号里
+	加
-	减
x	乘
÷	除
#	立即值
¨	被装入 （读：获得）
:	连接

条件寄存器 （CCR）位

V	溢出位
H	半进位位
I	中断屏蔽标识
N	负标识位
Z	零标识
C	进位 / 借位标识

寻址模式:

DIR	直接寻址模式
EXT	扩展寻址模式
IMM	立即寻址模式
INH	内在寻址模式
IX	无偏移量变址模式
IX 1	8 位偏移量变址模式
IX 2	16 位偏移量变址模式
IX+	无偏移量变址、变址加 1 寻址模式
IX1+	8 位偏移量变址、变址加 1 寻址模式
REL	相对寻址模式
SP1	8 位偏移量堆栈寻址模式
SP2	16 位偏移量堆栈寻址模式

循环码 :

f	自由周期。在自由周期， CPU 不需要使用系统总线，一个自由周期通常是系统总线时钟的一个周期并且通常是一个只读周期。
p	程序取回；从程序寄存器的下个连续地址读取。
r	读 8 位操作码
s	把一个字节推入 （写入）堆栈
u	从堆栈弹出 （读出）一个字节
v	从 0XFFxx 读矢量 (从高字节开始)
w	写 8 位操作码

CCR 常用符号 :

↕	设置或清零
–	不影响
U	未定义

表 7-3. Opcode Map (第 1 页，共 2 页)

Bit-Manipulation		Branch	Read-Modify-Write					Control		Register/Memory					
00 5 BRSET0 3 DIR	10 5 BSET0 2 DIR	20 3 BRA 2 REL	30 5 NEG 2 DIR	40 1 NEGA 1 INH	50 1 NEGX 1 INH	60 5 NEG 2 IX1	70 4 NEG 1 IX	80 9 RTI 1 INH	90 3 BGE 2 REL	A0 2 SUB 2 IMM	B0 3 SUB 2 DIR	C0 4 SUB 3 EXT	D0 4 SUB 3 IX2	E0 3 SUB 2 IX1	F0 3 SUB 1 IX
01 5 BRCLR0 3 DIR	11 5 BCLR0 2 DIR	21 3 BRN 2 REL	31 5 CBEQ 3 DIR	41 4 CBEQA 3 IMM	51 4 CBEQX 3 IMM	61 5 CBEQ 3 IX1+	71 5 CBEQ 2 IX+	81 6 RTS 1 INH	91 3 BLT 2 REL	A1 2 CMP 2 IMM	B1 3 CMP 2 DIR	C1 4 CMP 3 EXT	D1 4 CMP 3 IX2	E1 3 CMP 2 IX1	F1 3 CMP 1 IX
02 5 BRSET1 3 DIR	12 5 BSET1 2 DIR	22 3 BHI 2 REL	32 5 LDHX 3 EXT	42 5 MUL 1 INH	52 6 DIV 1 INH	62 1 NSA 1 INH	72 1 DAA 1 INH	82 5+ BGND 1 INH	92 3 BGT 2 REL	A2 2 SBC 2 IMM	B2 3 SBC 2 DIR	C2 4 SBC 3 EXT	D2 4 SBC 3 IX2	E2 3 SBC 2 IX1	F2 3 SBC 1 IX
03 5 BRCLR1 3 DIR	13 5 BCLR1 2 DIR	23 3 BLS 2 REL	33 5 COM 2 DIR	43 1 COMA 1 INH	53 1 COMX 1 INH	63 5 COM 2 IX1	73 4 COM 1 IX	83 11 SWI 1 INH	93 3 BLE 2 REL	A3 2 CPX 2 IMM	B3 3 CPX 2 DIR	C3 4 CPX 3 EXT	D3 4 CPX 3 IX2	E3 3 CPX 2 IX1	F3 3 CPX 1 IX
04 5 BRSET2 3 DIR	14 5 BSET2 2 DIR	24 3 BCC 2 REL	34 5 LSR 2 DIR	44 1 LSRA 1 INH	54 1 LSRX 1 INH	64 5 LSR 2 IX1	74 4 LSR 1 IX	84 1 TAP 1 INH	94 2 TXS 1 INH	A4 2 AND 2 IMM	B4 3 AND 2 DIR	C4 4 AND 3 EXT	D4 4 AND 3 IX2	E4 3 AND 2 IX1	F4 3 AND 1 IX
05 5 BRCLR2 3 DIR	15 5 BCLR2 2 DIR	25 3 BCS 2 REL	35 4 STHX 2 DIR	45 3 LDHX 3 IMM	55 4 LDHX 2 DIR	65 3 CPHX 3 IMM	75 5 CPHX 2 DIR	85 1 TPA 1 INH	95 2 TSX 1 INH	A5 2 BIT 2 IMM	B5 3 BIT 2 DIR	C5 4 BIT 3 EXT	D5 4 BIT 3 IX2	E5 3 BIT 2 IX1	F5 3 BIT 1 IX
06 5 BRSET3 3 DIR	16 5 BSET3 2 DIR	26 3 BNE 2 REL	36 5 ROR 2 DIR	46 1 RORA 1 INH	56 1 RORX 1 INH	66 5 ROR 2 IX1	76 4 ROR 1 IX	86 3 PULA 1 INH	96 5 STHX 3 EXT	A6 2 LDA 2 IMM	B6 3 LDA 2 DIR	C6 4 LDA 3 EXT	D6 4 LDA 3 IX2	E6 3 LDA 2 IX1	F6 3 LDA 1 IX
07 5 BRCLR3 3 DIR	17 5 BCLR3 2 DIR	27 3 BEQ 2 REL	37 5 ASR 2 DIR	47 1 ASRA 1 INH	57 1 ASRX 1 INH	67 5 ASR 2 IX1	77 4 ASR 1 IX	87 2 PSHA 1 INH	97 1 TAX 1 INH	A7 2 AIS 2 IMM	B7 3 STA 2 DIR	C7 4 STA 3 EXT	D7 4 STA 3 IX2	E7 3 STA 2 IX1	F7 2 STA 1 IX
08 5 BRSET4 3 DIR	18 5 BSET4 2 DIR	28 3 BHCC 2 REL	38 5 LSL 2 DIR	48 1 LSLA 1 INH	58 1 LSLX 1 INH	68 5 LSL 2 IX1	78 4 LSL 1 IX	88 3 PULX 1 INH	98 1 CLC 1 INH	A8 2 EOR 2 IMM	B8 3 EOR 2 DIR	C8 4 EOR 3 EXT	D8 4 EOR 3 IX2	E8 3 EOR 2 IX1	F8 3 EOR 1 IX
09 5 BRCLR4 3 DIR	19 5 BCLR4 2 DIR	29 3 BHCS 2 REL	39 5 ROL 2 DIR	49 1 ROLA 1 INH	59 1 ROLX 1 INH	69 5 ROL 2 IX1	79 4 ROL 1 IX	89 2 PSHX 1 INH	99 1 SEC 1 INH	A9 2 ADC 2 IMM	B9 3 ADC 2 DIR	C9 4 ADC 3 EXT	D9 4 ADC 3 IX2	E9 3 ADC 2 IX1	F9 3 ADC 1 IX
0A 5 BRSET5 3 DIR	1A 5 BSET5 2 DIR	2A 3 BPL 2 REL	3A 5 DEC 2 DIR	4A 1 DECA 1 INH	5A 1 DECX 1 INH	6A 5 DEC 2 IX1	7A 4 DEC 1 IX	8A 3 PULH 1 INH	9A 1 CLI 1 INH	AA 2 ORA 2 IMM	BA 3 ORA 2 DIR	CA 4 ORA 3 EXT	DA 4 ORA 3 IX2	EA 3 ORA 2 IX1	FA 3 ORA 1 IX
0B 5 BRCLR5 3 DIR	1B 5 BCLR5 2 DIR	2B 3 BMI 2 REL	3B 7 DBNZ 3 DIR	4B 4 DBNZA 2 INH	5B 4 DBNZX 2 INH	6B 7 DBNZ 3 IX1	7B 6 DBNZ 2 IX	8B 2 PSHH 1 INH	9B 1 SEI 1 INH	AB 2 ADD 2 IMM	BB 3 ADD 2 DIR	CB 4 ADD 3 EXT	DB 4 ADD 3 IX2	EB 3 ADD 2 IX1	FB 3 ADD 1 IX
0C 5 BRSET6 3 DIR	1C 5 BSET6 2 DIR	2C 3 BMC 2 REL	3C 5 INC 2 DIR	4C 1 INCA 1 INH	5C 1 INCX 1 INH	6C 5 INC 2 IX1	7C 4 INC 1 IX	8C 1 CLRH 1 INH	9C 1 RSP 1 INH		BC 3 JMP 2 DIR	CC 4 JMP 3 EXT	DC 4 JMP 3 IX2	EC 3 JMP 2 IX1	FC 3 JMP 1 IX
0D 5 BRCLR6 3 DIR	1D 5 BCLR6 2 DIR	2D 3 BMS 2 REL	3D 4 TST 2 DIR	4D 1 TSTA 1 INH	5D 1 TSTX 1 INH	6D 4 TST 2 IX1	7D 3 TST 1 IX		9D 1 NOP 1 INH	AD 5 BSR 2 REL	BD 5 JSR 2 DIR	CD 6 JSR 3 EXT	DD 6 JSR 3 IX2	ED 5 JSR 2 IX1	FD 5 JSR 1 IX
0E 5 BRSET7 3 DIR	1E 5 BSET7 2 DIR	2E 3 BIL 2 REL	3E 6 CPHX 3 EXT	4E 5 MOV 3 DD	5E 5 MOV 2 DIX+	6E 4 MOV 3 IMD	7E 5 MOV 2 IX+D	8E 2+ STOP 1 INH	9E Page 2	AE 2 LDX 2 IMM	BE 3 LDX 2 DIR	CE 4 LDX 3 EXT	DE 4 LDX 3 IX2	EE 3 LDX 2 IX1	FE 3 LDX 1 IX
0F 5 BRCLR7 3 DIR	1F 5 BCLR7 2 DIR	2F 3 BIH 2 REL	3F 5 CLR 2 DIR	4F 1 CLRA 1 INH	5F 1 CLRX 1 INH	6F 5 CLR 2 IX1	7F 4 CLR 1 IX	8F 2+ WAIT 1 INH	9F 1 TXA 1 INH	AF 2 AIX 2 IMM	BF 3 STX 2 DIR	CF 4 STX 3 EXT	DF 4 STX 3 IX2	EF 3 STX 2 IX1	FF 2 STX 1 IX

INH 隐含寻址
IMM 立即寻址
DIR 直接寻址
EXT 扩展寻址
DD 直接 – 直接寻址
DIX+ 直接 - 自动加 1 变址寻址
REL 相对寻址
IX 无偏移量变址寻址
IX1 8 位偏移量变址寻址
IX2 16 位偏移量变址寻址
IMD 立即 - 直接寻址
SP1 8 位偏移量堆栈寻址
SP2 16 位偏移量堆栈寻址
IX+ 无偏移量自动加 1 变址寻址
IX+D 变址 – 直接 – 变址加 1 寻址
IX1+ 8 位偏移量自动加 1 变址寻址

用 16 进制表示操作码 F0 3 **HCS08 指令周期**
SUB
字节数 1 IX **助记寻址方式**

表 7-3. Opcode Map (第 2 页，共 2 页)

Bit-Manipulation		Branch	Read-Modify-Write					Control		Register/Memory					
						9E60 6 NEG 3 SP1							9ED0 5 SUB 4 SP2	9EE0 4 SUB 3 SP1	
						9E61 6 CBEQ 4 SP1							9ED1 5 CMP 4 SP2	9EE1 4 CMP 3 SP1	
													9ED2 5 SBC 4 SP2	9EE2 4 SBC 3 SP1	
						9E63 6 COM 3 SP1							9ED3 5 CPX 4 SP2	9EE3 4 CPX 3 SP1	9EF3 6 CPHX 3 SP1
						9E64 6 LSR 3 SP1							9ED4 5 AND 4 SP2	9EE4 4 AND 3 SP1	
													9ED5 5 BIT 4 SP2	9EE5 4 BIT 3 SP1	
						9E66 6 ROR 3 SP1							9ED6 5 LDA 4 SP2	9EE6 4 LDA 3 SP1	
						9E67 6 ASR 3 SP1							9ED7 5 STA 4 SP2	9EE7 4 STA 3 SP1	
						9E68 6 LSL 3 SP1							9ED8 5 EOR 4 SP2	9EE8 4 EOR 3 SP1	
						9E69 6 ROL 3 SP1							9ED9 5 ADC 4 SP2	9EE9 4 ADC 3 SP1	
						9E6A 6 DEC 3 SP1							9EDA 5 ORA 4 SP2	9EEA 4 ORA 3 SP1	
						9E6B 8 DBNZ 4 SP1							9EDB 5 ADD 4 SP2	9EEB 4 ADD 3 SP1	
						9E6C 6 INC 3 SP1									
						9E6D 5 TST 3 SP1									
										9EAE 5 LDHX 2 IX	9EBE 6 LDHX 4 IX2	9ECE 5 LDHX 3 IX1	9EDE 5 LDX 4 SP2	9EEE 4 LDX 3 SP1	9EFE 5 LDHX 3 SP1
						9E6F 6 CLR 3 SP1							9EDF 5 STX 4 SP2	9EEF 4 STX 3 SP1	9EFF 5 STHX 3 SP1

INH	隐含寻址	**REL**	相对寻址	**SP1**	8 位偏移量堆栈寻址
IMM	立即寻址	**IX**	无偏移量变址寻址	**SP2**	16 位偏移量堆栈寻址
DIR	直接寻址	**IX1**	8 位偏移量变址寻址	**IX+**	无偏移量自动加 1 变址寻址
EXT	扩展寻址	**IX2**	16 位偏移量变址寻址	**IX+D**	变址 - 直接 - 变址加 1 寻址
DD	直接 - 直接寻址	**IMD**	立即 - 直接寻址	**IX1+**	8 位偏移量自动加 1 变址寻址
DIX+	直接 - 自动加 1 变址寻址				

用 16 进制表示操作码 和前置位 （9E）	9E60 6	HCS08 指令周期
	NEG	
字节数	3 SPI	助记寻址方式

第 8 章
循环冗余校验 (S08CRCV1)

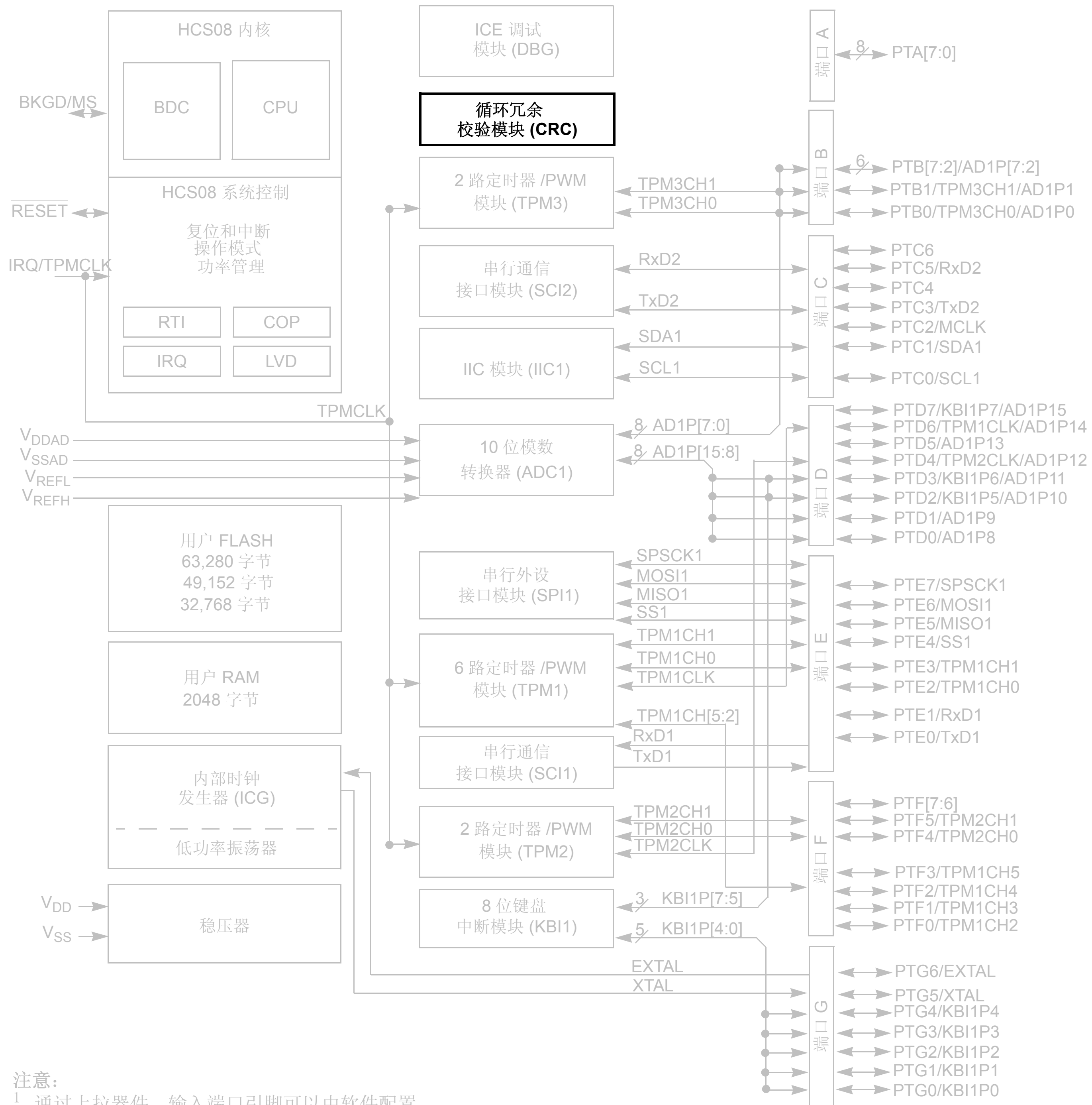
8.1 介绍

MC9S08AC60 系列包含 CRC 模块，支持存储器上的快速循环冗余校验。

8.1.1 特性

CRC 模块具有以下特性：

- 采用 16 位移位寄存器的硬件 CRC 发生器电路
- CRC16-CCITT 符合 $x^{16} + x^{12} + x^5 + 1$ 多项式
- 误码检测功能可以检测所有单、双、奇误码及大多数多位误码
- 可编程的初始速率值
- 高速 CRC 计算



注意：

[1] 通过上拉器件，输入端口引脚可以由软件配置。

[2] 如果 IRQ 使能 (IRQPE = 1)，引脚包含软件可配置的上拉 / 下拉器件。如果选择上升边沿检测，则下拉使能 (IRQEDG = 1)。

[3] 引脚包含集成上拉器件。

[4] PTD3、PTD2、PTD7 和 PTG4 既包含上拉也包含下拉器件。当 KBI 使能时 (KBIPEn = 1)，且选择了上升边沿 (KBEDGn = 1)，则下拉使能 。

[5] TPMCLK、 TPM1CLK 和 TPM2CLK 选项可通过软件配置； 复位后， TPM1CLK、 TPM2CLK 和 TPMCLK 分别提供给 TPM1、 TPM2 和 TPM3。

图 8-1. 显亮 CRC 模块的 MC9S08AC60 系列结构图

8.1.2 操作模式

本小节定义了 CRC 在运行、等待和停止模式下的操作。

- 运行模式 - 此为基本的操作模式。
- 等待模式 -CRC 模块正常运行。
- 停止 1 和停止 2 模式 - 这些模式下 CRC 不工作，从停止模式恢复后将进入复位状态。
- 停止 3 模式 - 在该模式下，CRC 模块将进入低功率待机状态。进行中的 CRC 计算将停止，等 CPU 进入运行模式后再继续计算。

8.1.3 结构图

图 8-2 为 CRC 模块的结构图。

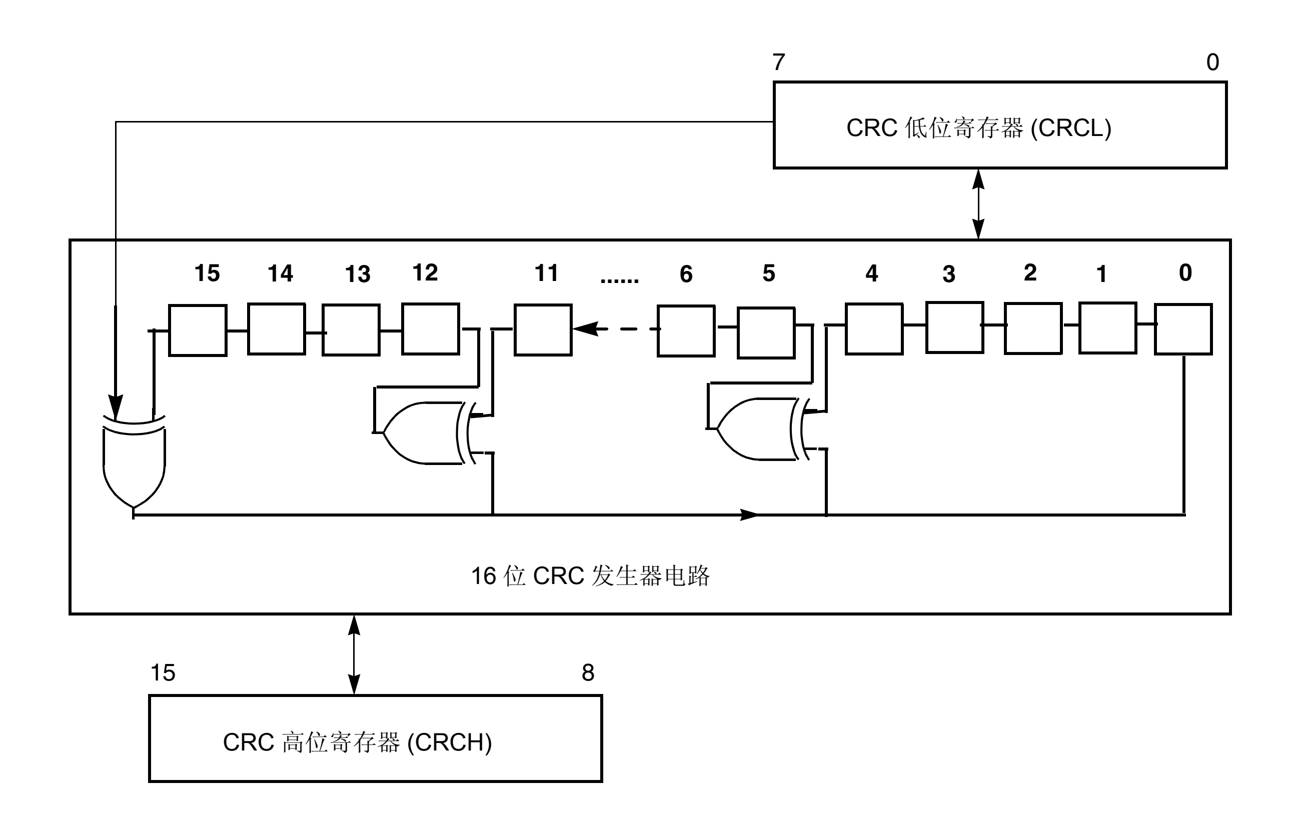


图 8-2. 循环冗余校验 （CRC）模块结构图

8.2 外部信号描述

CRC 信号无片外连接。

8.3 寄存器定义

8.3.1 存储器映射

表 8-1. CRC 寄存器一览

名称		7	6	5	4	3	2	1	0
CRCH	R	位 15	位 14	位 13	位 12	位 11	位 10	位 9	位 8
	W								
CRCL	R	位 7	位 6	位 5	ŒªÔ4	ŒªÔ3	ŒªÔ2	ŒªÔ1	ŒªÔ0
	W								

8.3.2 寄存器描述

CRC 模块包括 :

- 一个 16 位 CRC 结果和种子寄存器 (CRCH:CRCL)

有关所有 CRC 寄存器的绝对地址分配，请参见本数据手册存储器章节中的直接页面寄存器一览。本小节只按照名称列举了寄存器。可以用飞思卡尔提供的等同或标头文件把这些名称转换为相应的绝对地址。

8.3.2.1 CRC 高位寄存器 (CRCH)

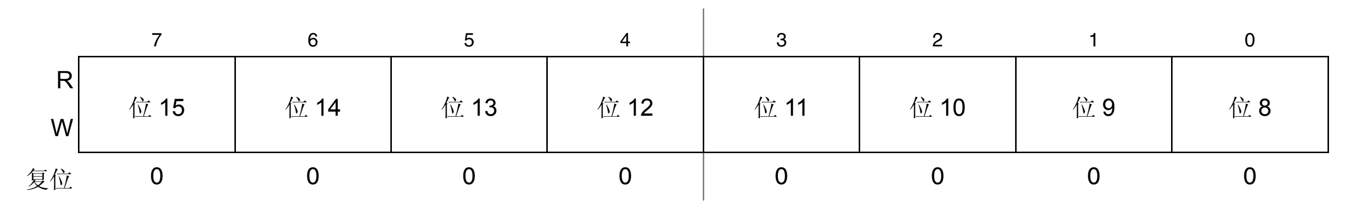


图 8-3. CRC 高位寄存器 (CRCH)

表 8-2. 寄存器字段描述

字段	描述
7:0 CRCH	**CRCH** -- 这是 16 位 CRC 寄存器的高位字节。写入 CRCH 将把初始 16 位种子值的高位字节直接加载到 CRC 发生器中移位寄存器的 15-8 位。CRC 发生器然后把低位字节写入到 CRCL 的种子值，并直接加载到 CRC 发生器中移位寄存器的 7-0 位。一旦写入到 CRCH:CRCL 的种子字节都加载到 CRC 发生器，且数据的一个字节已写入 CRCL，移位寄存器将开始移位。读取 CRCH 操作将从 CRC 发生器的移位寄存器直接读取当前 CRC 计算结果的 15-8 位。

8.3.2.2 CRC 低位寄存器 (CRCL)

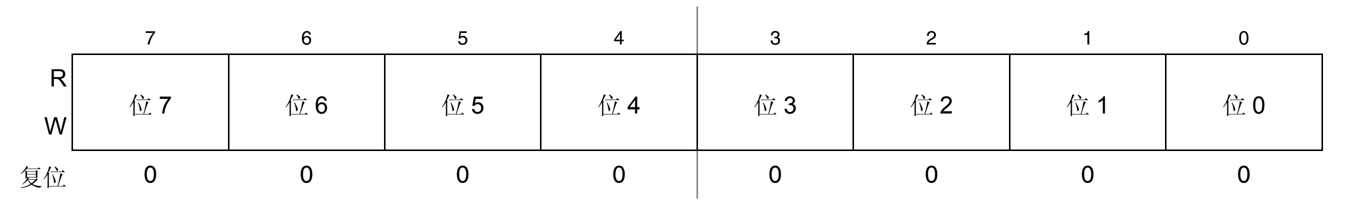


图 8-4. CRC 高位寄存器 (CRCH)

表 8-3. 寄存器字段描述

字段	描述
7:0 CRCL	**CRCL** -- 这是 16 位 CRC 寄存器的低位字节。通常情况下，写入 CRCL 将触发 CRC 发生器通过 16 位 CRC 发生器提供时钟。特殊情况下，如果之前已写入 CRCH，随后再进行的 CRCH 写入操作将把寄存器中的值作为 16 位种子值的低位字节直接加载到 CRC 发生器中移位寄存器的 7-0 位。读取 CRCL 操作将从 CRC 发生器的移位寄存器直接读取当前 CRC 计算结果的 7-0 位。

8.4 功能描述

使能 CRC 功能的步骤如下，对 CRCH 寄存器的写入操作将触发种子机制的前半部分，把 CRCH 数值直接加载到 CRC 发生器中移位寄存器的 15-8 位。然后， CRC 发生器将对 CRCL 进行写入操作，完成种子机制。

一旦 CRCL 寄存器被写入，它的值将被直接加载到移位寄存器的 7-0 位，完成种子机制的后半部分。在 CRCH:CRCL 中的值将是 CRC 发生器的初始种子值。

现在，要进行 CRC 计算的数据的第一个字节应写入到 CRCL。在种子机制完成之后的这一写入操作将触发 CRC 模块启动 CRC 校验进程。CRC 发生器将把 CRCL 寄存器中的位 (先是 MSB) 移动到发生器的移位寄存器内。在所有 8 个位都被移到 CRC 发生器之后 (在数据写入到 CRCL 后的下一个总线周期)，移位的结果，或者是目前在移位寄存器中的值，可以直接从 CRCH:CRCL 读取，而包含在 CRC 计算中的下一个数据字节可以写入 CRCL 寄存器。

然后，下一个字节也将通过 CRC 发生器的 16 位移位寄存器被移位，而在移位完成之后，第二次 CRC 计算的结果也能够直接从 CRCH:CRCL 读取。

在每个字节都完成移位之后，新的 CRC 结果将出现在 CRCH:CRCL 中，可以将另一个字节写入到 CRCL 寄存器 （包含在 CRC16-CCITT 计算中）。每当 8 位被移动到移位寄存器内时， CRCH:CRCL 中就会出现一个新的 CRC 结果。

要启动一个新 CRC 计算，则写入 CRCH，新 CRC 计算的种子机制将重新开始。

8.4.1 ITU-T(CCITT) 建议标准以及预期的 CRC 结果

CRC 多项式 0x1021 ($x^{16} + x^{12} + x^5 + 1$) 一般被称作 *CRC-CCITT*，因为它最初是由国际电信联盟 (ITU-T，前身为国际电报电话咨询委员会，即 CCITT) 委员会所建议使用的。

尽管国际电信联盟的建议清楚地定义了要使用的多项式 0x1021，但是他们也接受实施中的变通：

- ITU-T V.41 实施的电路与图 8-2 中所示的相同，但是它建议 SEED = 0x0000。
- ITU-T T.30 和 ITU-T X.25 实施的电路与图 8-2 中所示的相同，但是它们建议最后的 CRC 结果取非（反码操作）。而且，他们还建议 SEED = 0xFFFF。

此外，从文献中经常发现与上述建议中略有不同的电路，但是它们也被称作是 CRC-CCITT 电路 (很多此类电路要求报文增零)。

CRC 模块中所采用的电路正是 ITU-T V.41 标准所建议的，而且，增加了灵活的可编程 SEED。与 ITU-T V.41 标准中的要求一样，没有必要进行增加数位的处理， CRC 结果不进行反码操作。下面是一些预期的结果，以资参考。注意这些是 ASCII 消息。例如， 123456789 被编码为 0x31 到 0x39 （见 ASCII 表）

表 8-4. 预期的 CRC 结果

报文	种子 (初始 CRC 值)	CRC 结果
A	0x0000	0x58e5
A	0xffff	0xb915
123456789	0x0000	0x31c3
123456789	0xffff	0x29b1
256 个大写字符 "A" 无中断	0x0000	0xabe3
256 个大写字符 "A" 无中断	0xffff	0xea0b

8.5 初始化信息

要初始化 CRC 模块，并启动 CRC16-CCITT 计算，请按以下步骤操作 :

1. 把初始种子值的高位字节写入 CRCH。
2. 把初始种子值的低位字节写入 CRCL。
3. 把 CRC 计算用的数据的第一个字节写入 CRCL。
4. 在步骤 3 之后的下一个总线周期，如果需要，第一个字节的 CRC 结果可以从 CRCH:CRCL 读取。
5. 重复第三和第四步，直到所有数据都校验完毕。

第 9 章
模数转换器 (S08ADC10V1)

9.1 概述

10 位模数转换器 （ADC）是新一代的逼近模数转换器，在集成的微处理器片上系统中运行。这种 ADC 模块设计支持最高 28 个独立的模拟输入 （AD0-AD27）。MC9S08AC60 系列微处理器上只使用了其中 18 个 （AD0-AD15、AD26 和 AD27）输入。这些输入通过 ADCH 位选择。一些输入与 I/O 引脚共享，参见 图 9-1。表 9-1 中总结了 MC9S08AC60 系列器件的所有 ADC 通道分配。

9.2 通道分配

MC9S08AC60 系列器件的 ADC 通道分配参见下表。未使用的通道在内部连接到 V_{REFL}。预留的通道转为未知值。连接到 I/O 引脚的通道有相关的引脚控制位，如下所示。

表 9-1. ADC 通道分配

ADCH	通道	输入	引脚控制	ADCH	通道	输入	引脚控制
00000	AD0	PTB0/ADCP0	ADPC0	10000	AD16	V_{REFL}	N/A
00001	AD1	PTB1/ADCP1	ADPC1	10001	AD17	V_{REFL}	N/A
00010	AD2	PTB2/ADCP2	ADPC2	10010	AD18	V_{REFL}	N/A
00011	AD3	PTB3/ADCP3	ADPC3	10011	AD19	V_{REFL}	N/A
00100	AD4	PTB4/ADCP4	ADPC4	10100	AD20	V_{REFL}	N/A
00101	AD5	PTB5/ADCP5	ADPC5	10101	AD21	V_{REFL}	N/A
00110	AD6	PTB6/ADCP6	ADPC6	10110	AD22	Reserved	N/A
00111	AD7	PTB7/ADCP7	ADPC7	10111	AD23	Reserved	N/A
01000	AD8	PTD0/ADCP8	ADPC8	11000	AD24	Reserved	N/A
01001	AD9	PTD1/ADCP9	ADPC9	11001	AD25	Reserved	N/A
01010	AD10	PTD2/ADCP10/ KBI1P5	ADPC10	11010	AD26	Temperature Sensor[1]	N/A
01011	AD11	PTD3/ADCP11/ KBI1P6	ADPC11	11011	AD27	Internal Bandgap[2]	N/A
01100	AD12	PTD4/ADCP12/ TPM2CLK	ADPC12	11100	-	Reserved	N/A
01101	AD13	PTD5/ADCP13	ADPC13	11101	V_{REFH}	V_{REFH}	N/A
01110	AD14	PTD6/ADCP14/ TPM1CLK	ADPC14	11110	V_{REFL}	V_{REFL}	N/A
01111	AD15	PTD7/ADCP15/ KBI1P7	ADPC15	11111	module disabled	None	N/A

[1] 更多信息，请参见 9.2.3 节 温度传感器。

[2] 选择内部带隙通道要求 SPMSC1 中 BGBE =1，参见 5.9.8 节 系统电源管理状态和控制寄存器 1 (SPMSC1)。带隙电压参考的值，参见 A.6 节 DC 特性。

9.2.1 替代时钟

ADC 模块可采用 MCU 总线时钟执行转换，总线时钟分为两个，即模块中的本地异步时钟 （ ADACK） 和替代时钟， ALTCLK。MC9S08AC60 系列 MCU 器件的替代时钟是内部时钟发生器 （ICG）模块的外部参考时钟 （ICGERCLK）。

由于只有当外部时钟源使能时，ICGERCLK 才激活，所以 ICG 必须配置为 FBE 或 FEE 模式 （CLKS1 = 1）。 ICGERCLK 运行的频率必须使 ADC 转换时钟在由 ADIV 位确定的从 ALTCLK 输入分频后能够在其规定的频率范围 (f_{ADCK}) 内运行 。例如，如果 ADIV 位设置为除以四，则 ALTCLK (ICGERCLK) 的最小频率是 f_{ADCK} 最小值的四倍，其最大频率是 f_{ADCK} 最大值的四倍。由于有最低频率要求，因此当使用振荡器电路时，它必须配置为高量程运行 （RANGE = 1）。

如果上述条件满足，当 MCU 处于等待模式时， ALTCLK 激活。这样，当 MCU 处于等待模式时， ALTCLK 可以用作 ADC 的转换时钟源。

当 MCU 处于停止 3 模式时， ALTCLK 不能用作 ADC 转换时钟源。

9.2.2 硬件触发

ADC 硬件触发 ADHWT 是实时中断 (RTI) 计数器的输出。RTI 计数器可以由 ICGERCLK 或 RTI 块中的标准 1 kHz 时钟源提供时钟。 MCU 处于运行、等待或停止 3 模式时， 1 kHz 时钟源都可以使用。如果 ICG 配置为 FBE 或 FEE 模式，MCU 在运行或等待模式时， ICGERCLK 可以使用。

RTI 的周期由输入时钟频率和 RTIS 位决定。当 ADC 硬件触发使能时，在 RTI 计数器溢出时启动转换。 RTI 计数器是自由运行计数器，以 RTI 速率生成溢出， RTI 速率由 RTIS 位决定。

注意

ADC 触发在第一个 RTI 溢出时产生，以后每两个 RTI 计数器溢出时产生一次。这是因为 RTI 计数器会到期，且 ADC 触发是在 RTI 输出的上升边沿发生。

9.2.2.1 模拟引脚使能

MC9S08AC60 系列 上的 ADC 只包含两个模拟引脚使能寄存器， APCTL1 和 APCTL2。

9.2.2.2 低功率模式操作

ADC 能够在停止 3 模式中运行，但是要在 SPMSC1 中设置 LVDSE 和 LVDE。

9.2.3 温度传感器

ADC 模块包括一个温度传感器，这个传感器的输出连接到 ADC 模拟通道输入。公式 9-1 给出了温度传感器的逼近转移函数。

$$\text{Temp} = 25 - ((V_{TEMP} - V_{TEMP25}) \div m) \qquad \textit{公式 9-1}$$

其中：

- V_{TEMP} 是室温时温度传感器通道的电压。
- V_{TEMP25} 是 25°C 时温度传感器通道的电压。
- m 是冷热压与温度的比值，单位为 V/°C。

计算温度时，从 ADC 电气表中获取 V_{TEMP25} 和 m 值。

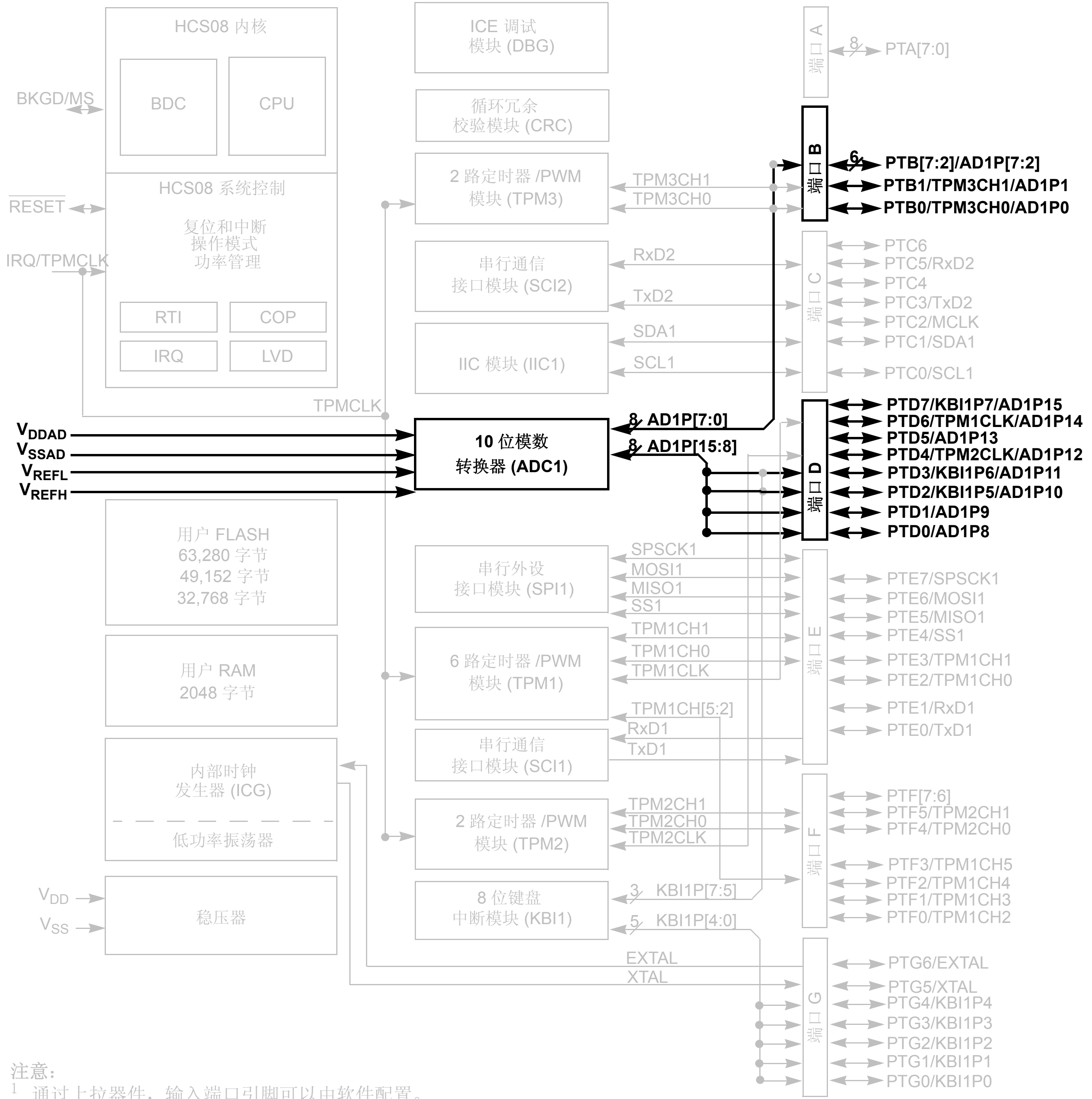
在应用代码中，用户读取温度传感器通道，计算 V_{TEMP}，并与 V_{TEMP25} 进行比较。如果 V_{TEMP} 大于 V_{TEMP25}，则公式 9-1 应用冷斜率值。如果 V_{TEMP} 低于 V_{TEMP25}，则公式 9-1 应用热斜率值。

要提高准确率，需要校准带隙电压参考和温度传感器。

25 °C 进行的校准会使准确率提高 ± 4.5 °C。

-40 °C, 25 °C 和 125 °C 这三个温度点进行的校准可以将准确率提高 ± 2.5 °C。一旦完成校准，用户需要校准热压和冷压的斜率。在应用代码中，用户则用公式 9-1 计算温度，参见上面的描述，然后判断温度是否高于或低于 25 °C。确定温度是高于或是低于 25 °C 后，用户可以用在校准过程中获得的冷或热斜率重新计算温度。

有关使用温度传感器的更多信息，请参考 AN3031。



注意：

[1] 通过上拉器件，输入端口引脚可以由软件配置。

[2] 如果 IRQ 使能 (IRQPE = 1)，引脚包含软件可配置的上拉 / 下拉器件。如果选择上升边沿检测，则下拉使能 (IRQEDG = 1)。

[3] 引脚包含集成上拉器件。

[4] PTD3、PTD2、PTD7 和 PTG4 既包含上拉也包含下拉器件。当 KBI 使能时 (KBIPEn = 1)，且选择了上升边沿 (KBEDGn = 1)，则下拉使能 。

[5] TPMCLK、 TPM1CLK 和 TPM2CLK 选项可通过软件配置； 复位后， TPM1CLK、 TPM2CLK 和 TPMCLK 分别提供给 TPM1、 TPM2 和 TPM3。

图 9-1. 显亮 ADC 块和引脚的 MC9S08AC60 结构图

9.2.4 特点

ADC 模块特点包括：

- 线性逐次逼近算法， 10 位精度。
- 多达 28 个模拟输入。
- 8 位或 10 位右对齐格式输出
- 单个或连续的转换 （单个转换后自动返回到空闲）
- 设置采样时间和转换速度 （功率）
- 转换完成标志和中断
- 输入时钟可以选择高达四个时钟源
- 在等待或 stop3 模式中操作为低噪音操作
- 异步时钟源的低噪音操作
- 可选的异步硬件转换触发
- 自动比较小于，大于或等于编程值
- 温度传感器

9.2.5 框图

图 9-2 提供了 ADC 模块的框图。

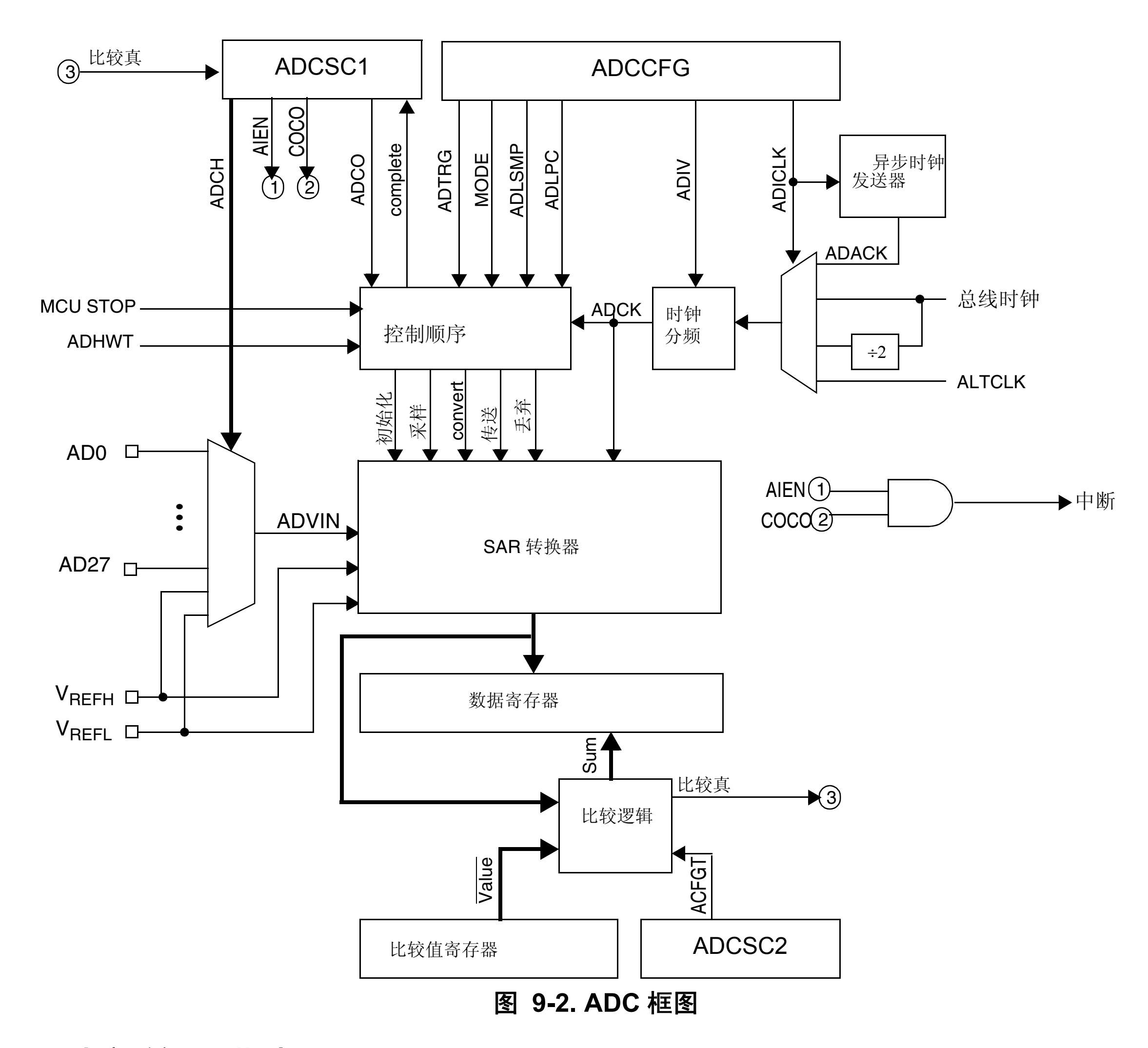


图 9-2. ADC 框图

9.3 外部信号描述

ADC 模块支持高达 28 个独立模拟输入。也需要 4 个电源 / 参考 / 地连接。

表 9-2. 信号属性

名称	功能
AD27-AD0	模拟通道输入
V_{REFH}	高参考电压
V_{REFL}	低参考电压
V_{DDAD}	模拟电压供电
V_{SSAD}	模拟地

9.3.1 模拟电源 （V_{DDAD}）

ADC 模拟部分使用作为它的电源连接。在相同的封装中， V_{DDAD} 在内部连接到 V_{DD}。如果外部可能，连接到 V_{DDAD} 的引脚和 V_{DD} 到相同电压。外部滤波对 V_{DDAD} 可能是必要的。

9.3.2 模拟地 （V_{SSAD}）

ADC 模拟部分使用作为它的地连接。在相同的封装中， V_{SSAD} 在内部连接到 V_{SS}。如果外部可能，连接到 V_{SSAD} 引脚和 V_{SS} 到相同电压。

9.3.3 参考高电压 （V_{REFH}）

V_{REFH} 是转换器的参考高电压，在相同的封装中， V_{REFH} 在内部连接到 V_{DDAD}，如果外部可能，连接到 V_{REFH} 的引脚和 V_{DDAD} 到相同电压。或者被外部源 （在最小的 V_{DDAD} 规格和 V_{DDAD} 电压之间）驱动。

9.3.4 参考低电压 （V_{REFL}）

V_{REFL} 是转换器的参考低电压，在相同的封装中， V_{REFL} 在内部连接到 V_{SSAD}，如果外部可能，连接到 V_{REFL} 的引脚和 V_{SSAD} 到相同电压。

9.3.5 模拟通道输入 （ADx）

ADC 模块支持高达 28 个独立的模拟输入。通过 ADCH 通道选择位，一个输入被选择用于转换。

9.4 寄存器定义

这些内存映像寄存器控制和管理 ADC 的操作：

- 状态和控制寄存器， ADCSC1
- 状态和控制寄存器， ADCSC2
- 数据结果寄存器， ADCRH 和 ADCRL
- 比较值寄存器， ADCCVH 和 ADCCVL
- 配置寄存器， ADCCFG
- 引脚使能寄存器， APCTL1、 APCTL2、 APCTL3

9.4.1 状态和控制寄存器 1 （ADCSC1）

本节描述 ADC 状态和控制寄存器 （ADCSC1）的功能。写 ADCSC1 可以终止当前的转换并初始化一个新的转换 （如果 ADCH 等于一个不是全 1 的值）。

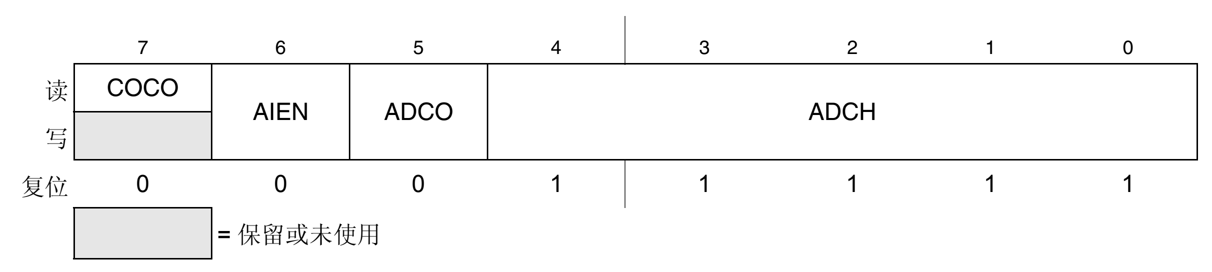


图 9-3. 状态和控制寄存器 (ADCSC1)

表 9-3. ADCSC1 寄存器域描述

域	描述
7 COCO	转换完成标志——COCO 标志是一个只读位。当比较功能禁止 （ACFE=0）时，每次转换完成时置位。当比较功能允许 （ACFE=1）时，转换完成后，只要比较结果为真，则 COCO 置位。只要写 ADCSC1 或读 ADCRL，该位清零。 0 转换未完成。 1 转换完成。
6 AIEN	中断允许——AIEN 用于使能转换完成中断。当 AIEN 为高， COCO 置位时，确认一个中断。 0 禁止转换完成中断。 1 允许转换完成中断。
5 ADCO	连续转换使能——ADCO 用于使能连续转换。 0 当选择软件触发中断时，写 ADCSC1 后开始一个转换。当选择硬件触发中断时，确认了 ADHWT 后开始一个中断。 1 当选择软件触发中断时，写 ADCSC1 后初始化连续转换。当选择硬件触发中断时，连续转换被 ADHWT 事件初始化。
4:0 ADCH	输入通道选择——ADCH 包括 5 位，用于选择输入通道中的一个。输入通道在表 9-4 中描述。 当通道选择位设置为全 1 时，逐次逼近转换器子系统关闭。这个特点允许禁止 ADC 和从所有的源中孤立输入通道。

表 9-4. 输入通道选择

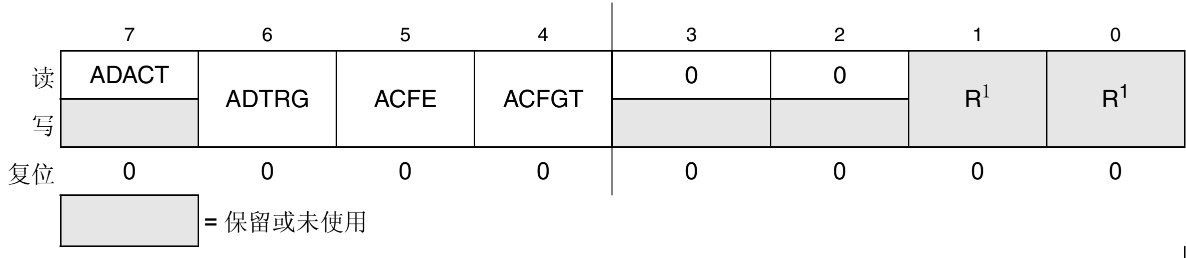
ADCH	输入选择
00000	AD0
00001	AD1
00010	AD2
00011	AD3
00100	AD4
00101	AD5
00110	AD6
00111	AD7
01000	AD8
01001	AD9
01010	AD10
01011	AD11
01100	AD12
01101	AD13
01110	AD14
01111	AD15
10000	AD16
10001	AD17

表 9-4. 输入通道选择 (续)

ADCH	输入选择
10010	AD18
10011	AD19
10100	AD20
10101	AD21
10110	AD22
10111	AD23
11000	AD24
11001	AD25
11010	AD26
11011	AD27
11100	保留
11101	V_{REFH}
11110	V_{REFL}
11111	模块被禁止

9.4.2 状态和控制寄存器 2 （ADCSC2）

ADCSC2 寄存器用于控制比较功能，转换触发和 ADC 模块的转换行为。



[1] 位 1 和位 0 为保留位，必须写为 0.

图 9-4. 状态和控制寄存器 (ADCSC2)

表 9-5. ADCSC2 寄存器域描述

域	描述
7 ADACT	转换行为——ADACT 表示转换正在进行中。当初始化转换时， ADACT 置位；当转换完成或终止时， ADACT 清零。 0 转换未进行。 1 转换处理中。
6 ADTRG	转换触发选择——ADTRG 用于选择初始化转换的触发的类型。两种触发类型可选：软件触发和硬件触发。选择软件触发，写 ADCSC1 后初始化一个转换。选择硬件触发，确认了 ADHWT 输入后初始化转换。 0 选择软件触发。 1 需安装硬件触发。

表 9-5. ADCSC2 寄存器域描述 （续）

5 ACFE	比较功能使能——ACFE 用于使能比较功能。 0 禁止比较功能。 1 允许比较功能。
4 ACFGT	比较功能更大使能——当监控的输入的转换结果大于或等于比较结果时， ACFGT 用于设置比较功能的触发。当监控的输入的转换结果小于比较结果时，比较功能默认触发。 0 当输入小于比较电平时，比较触发。 1 当输入大于或等于比较电平时，比较触发。

9.4.3 数据高结果寄存器 （ADCRH）

ADCRH 包含 10 位转换结果的高 2 位。当设置为 8 位转换时， ADR8 和 ADR9 都等于 0。每次转换完成，除非自动比较被允许而且不满足比较结果，ADCRH 将被更新，。在 10 位数据模式中，读 ADCRH 将暂时禁止下一次转换，直到读取了 ADCRL 中的内容。如果直到下一个转换完成都没有读 ADCRL，这个中间转换结果将会丢失。在 8 位数据模式中，没有对 ADCRL 的互锁。在这种情况下， MODE 位被改变， ADCRH 中的任何数据都无效。

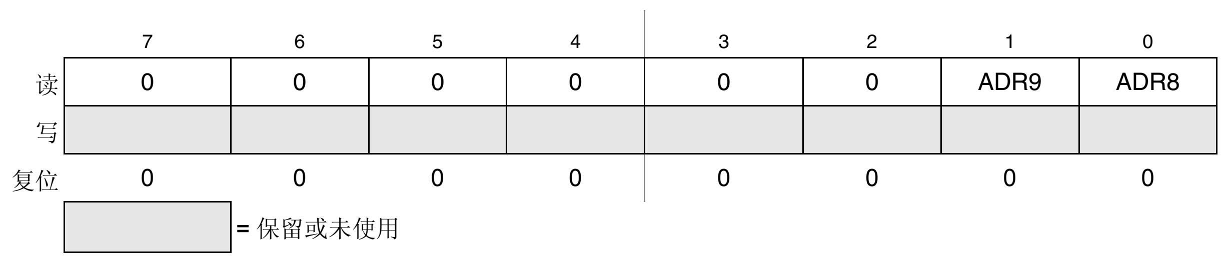


图 9-5. 数据高结果寄存器 (ADCRH)

9.4.4 数据低结果寄存器 （ADCRL）

ADCRL 包含 10 位转换结果的低 8 位，一个 8 位转换的所有 8 位。每次转换完成，这个寄存器都被更新，除非自动比较被允许而且不满足比较结果。在 10 位数据模式中，读 ADCRH 将暂时禁止下一次转换，直到读取了 ADCRL 中的内容。如果直到下一个转换完成都没有读 ADCRL，这个中间转换结果将会丢失。在 8 位数据模式中，没有对 ADCRH 的互锁。在这种情况下， MODE 位被改变， ADCRL 中的任何数据都无效。

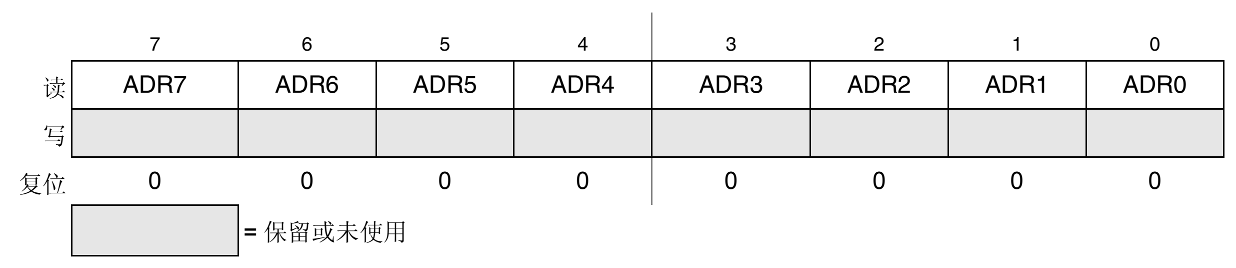


图 9-6. 数据低结果寄存器 (ADCRL)

9.4.5 比较值高寄存器 （ADCCVH）

该寄存器包含了 10 位比较值的高 2 位。当允许比较功能时，这些位和 10 位模式中的转换结果的高 2 位比较。在 8 位操作中， ADCCVH 在比较过程中不使用。

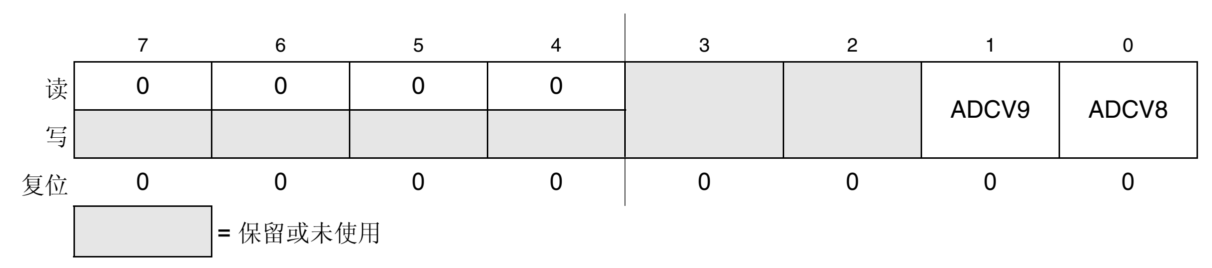


图 9-7. 比较值高寄存器 (ADCCVH)

9.4.6 比较值低寄存器 （ADCCVL）

该寄存器包含了 10 位比较值的低 8 位，或者 8 位比较值的所有 8 位。在 10 位或 8 位模式中，ADCV7:ADCV0 转换结果的低 8 位比较。

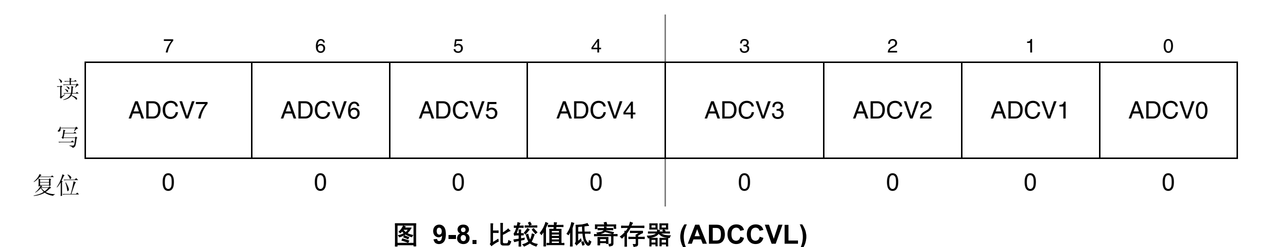


图 9-8. 比较值低寄存器 (ADCCVL)

9.4.7 配置寄存器 （ADCCFG）

ADCCFG 用于选择操作模式，时钟源，时钟分频和低功耗或长采样时间的设置。

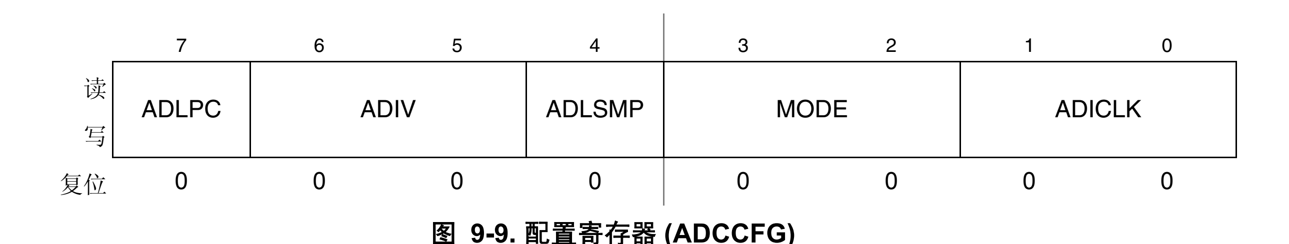


图 9-9. 配置寄存器 (ADCCFG)

表 9-6. ADCCFG 寄存器域描述

域	描述
7 ADLPC	低功耗配置——ADLPC 控制逐次渐进转换器的速度和功耗配置。当不需要更高采样速率时，可以优化功耗。 0 高速配置。 1 低功耗配置：以最大化时钟速率的代价降低功耗。
6:5 ADIV	时钟分频选择——ADIV 选择 ADC 使用的分频因子，产生内部时钟 ADCK。表 9-7 描述了时钟配置。
4 ADLSMP	长采样时间配置——ADLSMP 选择长和短采样时间。这可以调整采样周期，使在更高阻抗的输入下也能得到精确的采样，也可以在低阻抗时最大化转换速度。如果连续采样允许而且不需要高输出斜率，更长的采样时间可以用于更低的总功耗。 0 短采样时间。 1 长采样时间。

表 9-6. ADCCFG 寄存器域描述 （续）

3:2 MODE	转换模式选择——MODE 位用于选择 8 位或 10 位操作。参见表 9-8。
1:0 ADICLK	输入时钟选择——ADICLK 选择产生内部时钟 ADCK 的输入时钟源。参见表 9-9。

表 9-7. 时钟分频选择

ADIV	分频因子	时钟
00	1	输入时钟
01	2	输入时钟 /2
10	4	输入时钟 /4
11	8	输入时钟 /8

表 9-8. 转换模式

模式	模式描述
00	8 位转换 （N=8）
01	保留
10	10 位转换 （N=10）
11	保留

表 9-9. 输入时钟选择

ADICLK	时钟源选择
00	总线时钟
01	总线时钟 /2
10	交替时钟 (ALTCLK)
11	异步时钟 (ADACK)

9.4.8 引脚控制 1 寄存器 （APCTL1）

引脚控制寄存器用于禁止 MCU 引脚的 I/O 口用作模拟输入。 APCTL1 用于控制和 ADC 模块的通道 0-7 相关的位。

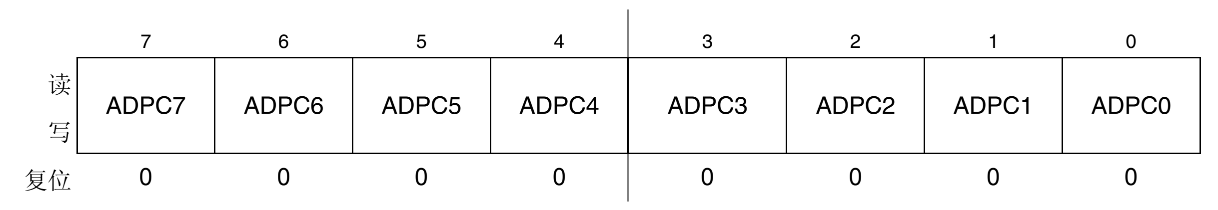


图 9-10. 引脚控制 1 寄存器 (APCTL1)

表 9-10. APCTL1 寄存器域描述

域	描述
7 ADPC7	ADC 引脚控制 7——ADPC7 用于控制和通道 AD7 相关的位。 0 允许 AD7 引脚 I/O 控制。 1 禁止 AD7 引脚 I/O 控制。
6 ADPC6	ADC 引脚控制 6——ADPC6 用于控制和通道 AD6 相关的位。 0 允许 AD6 引脚 I/O 控制。 1 禁止 AD6 引脚 I/O 控制。
5 ADPC5	ADC 引脚控制 5——ADPC5 用于控制和通道 AD5 相关的位。 0 允许 AD5 引脚 I/O 控制。 1 禁止 AD5 引脚 I/O 控制。
4 ADPC4	ADC 引脚控制 4——ADPC4 用于控制和通道 AD4 相关的位。 0 允许 AD4 引脚 I/O 控制。 1 禁止 AD4 引脚 I/O 控制。
3 ADPC3	ADC 引脚控制 3——ADPC3 用于控制和通道 AD3 相关的位。 0 允许 AD3 引脚 I/O 控制。 1 禁止 AD3 引脚 I/O 控制。
2 ADPC2	ADC 引脚控制 2——ADPC2 用于控制和通道 AD2 相关的位。 0 允许 AD2 引脚 I/O 控制。 1 禁止 AD2 引脚 I/O 控制。
1 ADPC1	ADC 引脚控制 1——ADPC1 用于控制和通道 AD1 相关的位。 0 允许 AD1 引脚 I/O 控制。 1 禁止 AD1 引脚 I/O 控制。
0 ADPC0	ADC 引脚控制 0——ADPC0 用于控制和通道 AD0 相关的位。 0 允许 AD0 引脚 I/O 控制。 1 禁止 AD0 引脚 I/O 控制。

9.4.9 引脚控制 2 寄存器 （APCTL2）

APCTL2 用于控制和 ADC 模块的通道 8-15 相关的位。

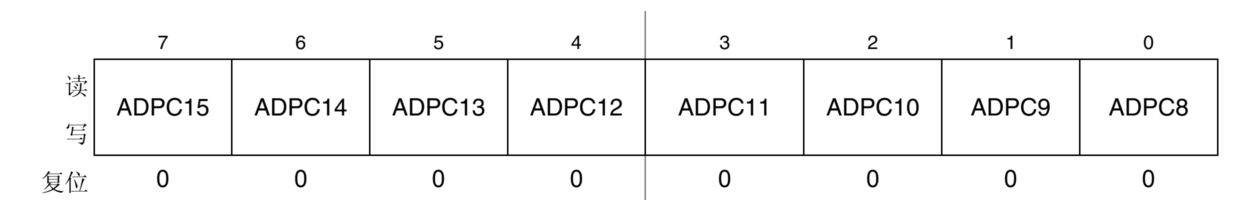


图 9-11. 引脚控制 2 寄存器 (APCTL2)

表 9-11. APCTL2 寄存器域描述

域	描述
7 ADPC15	ADC 引脚控制 15——ADPC15 用于控制和通道 AD15 相关的位。 0 允许 AD15 引脚 I/O 控制。 1 禁止 AD15 引脚 I/O 控制。

表 9-11. APCTL2 寄存器域描述 (续)

6 ADPC14	ADC 引脚控制 14——ADPC14 用于控制和通道 AD14 相关的位。 0 允许 AD14 引脚 I/O 控制。 1 禁止 AD14 引脚 I/O 控制。
5 ADPC13	ADC 引脚控制 13——ADPC13 用于控制和通道 AD13 相关的位。 0 允许 AD13 引脚 I/O 控制。 1 禁止 AD13 引脚 I/O 控制。
4 ADPC12	ADC 引脚控制 12——ADPC12 用于控制和通道 AD12 相关的位。 0 允许 AD12 引脚 I/O 控制。 1 禁止 AD12 引脚 I/O 控制。
3 ADPC11	ADC 引脚控制 11——ADPC11 用于控制和通道 AD11 相关的位。 0 允许 AD11 引脚 I/O 控制。 1 禁止 AD11 引脚 I/O 控制。
2 ADPC10	ADC 引脚控制 10——ADPC10 用于控制和通道 AD10 相关的位。 0 允许 AD10 引脚 I/O 控制。 1 禁止 AD10 引脚 I/O 控制。
1 ADPC9	ADC 引脚控制 9——ADPC9 用于控制和通道 AD9 相关的位。 0 允许 AD9 引脚 I/O 控制。 1 禁止 AD9 引脚 I/O 控制。
0 ADPC8	ADC 引脚控制 8——ADPC8 用于控制和通道 AD8 相关的位。 0 允许 AD8 引脚 I/O 控制。 1 禁止 AD8 引脚 I/O 控制。

9.4.10 引脚控制 3 寄存器 （APCTL3）

APCTL3 用于控制和 ADC 模块的通道 16-23 相关的位。

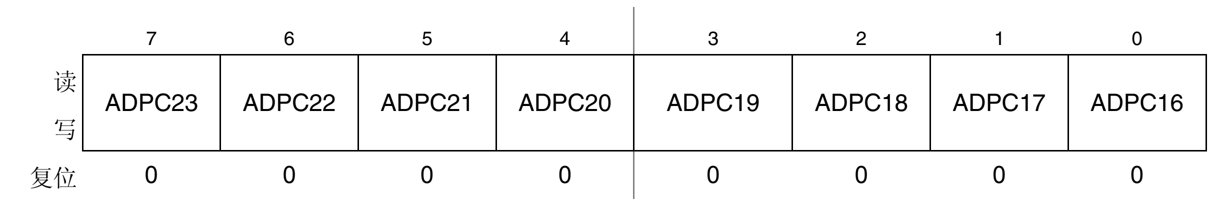


图 9-12. 引脚控制 3 寄存器 (APCTL3)

表 9-12. APCTL3 寄存器域描述

域	描述
7 ADPC23	ADC 引脚控制 23——ADPC23 用于控制和通道 AD23 相关的位。 0 允许 AD23 引脚 I/O 控制。 1 禁止 AD23 引脚 I/O 控制。
6 ADPC22	ADC 引脚控制 22——ADPC22 用于控制和通道 AD22 相关的位。 0 允许 AD22 引脚 I/O 控制。 1 禁止 AD22 引脚 I/O 控制。
5 ADPC21	ADC 引脚控制 21——ADPC21 用于控制和通道 AD21 相关的位。 0 允许 AD21 引脚 I/O 控制。 1 禁止 AD21 引脚 I/O 控制。

表 9-12. APCTL3 寄存器域描述 (续)

4 ADPC20	ADC 引脚控制 20——ADPC20 用于控制和通道 AD20 相关的位。 0 允许 AD20 引脚 I/O 控制。 1 禁止 AD20 引脚 I/O 控制。
3 ADPC19	ADC 引脚控制 19——ADPC19 用于控制和通道 AD19 相关的位。 0 允许 AD19 引脚 I/O 控制。 1 禁止 AD19 引脚 I/O 控制。
2 ADPC18	ADC 引脚控制 18——ADPC18 用于控制和通道 AD18 相关的位。 0 允许 AD18 引脚 I/O 控制。 1 禁止 AD18 引脚 I/O 控制。
1 ADPC17	ADC 引脚控制 17——ADPC17 用于控制和通道 AD17 相关的位。 0 允许 AD17 引脚 I/O 控制。 1 禁止 AD17 引脚 I/O 控制。
0 ADPC16	ADC 引脚控制 16——ADPC16 用于控制和通道 AD16 相关的位。 0 允许 AD16 引脚 I/O 控制。 1 禁止 AD16 引脚 I/O 控制。

9.5 功能描述

当复位或 ADCH 位全高时，禁止 ADC 模块。当转换完成而且另一个转换还未初始化时，该模块空闲。空闲时，模块处于最小功耗状态。

ADC 可以通过软件选择任何一个通道进行模数转换。选择的通道电压可以被逐次渐进算法转换成 11 位数字的结果。在 8 位模式中，选择的通道电压可以被逐次渐进算法转换成 9 位数字的结果。

当转换完成，结果放在数据寄存器中 （ADCRH 和 ADCRL）。在 10 位模式中，结果四舍五入成 10 位放在 ADCRH 和 ADCRL 中。在 8 位模式中，结果四舍五入成 8 位放在 ADCRL 中。转换完成标志置 1 并且如果允许转换完成中断 （AIEN=1），产生一个中断。

ADC 模块能够自动比较转换结果和比较寄存器的内容。通过置位 ACFE 位，允许比较功能。该功能和任何转换模式和设置协力完成操作。

9.5.1 时钟选择和分频控制

可以选择 4 个时钟源中的一个作为 ADC 模块的时钟源。这个时钟源除以一个设置值就可以产生转换器的输入时钟 （ADCK）。时钟选择下面源中的一个，由 ADICLK 位决定。

- 总线时钟，等于软件执行时的频率。这是复位后的默认值。
- 总线时钟 /2。对于更高的时钟，可以允许最大除以 16。
- ALTCLK，由 MCU 定义 （参见模块一节的介绍）
- 异步时钟 （ADACK）——该时钟由 ADC 模块内部的时钟源产生。当选择这个时钟源时，若 MCU 处于等待或 stop3 模式时，该时钟仍有效，允许在这些模式中以更低的噪音操作来进行转换。

无论选择哪个时钟，它的频率都必须低于规定的 ADCK 频率范围。如果可用的时钟太慢，根据规定 ADC 将不会工作。如果可用的时钟太快，时钟必须分频到适当的频率。分频因子由 ADIV 位决定，可以除以 1， 2，4， 8。

9.5.2 输入选择和引脚控制

引脚控制寄存器 （APCTL3、 APCTL2、 APCTL1）可以禁止引脚的 I/O 控制用于模拟输入。当引脚控制寄存器位置位时，相应的 MCU 位将会服从接下来的条件：

- 输出缓冲区强制为高阻抗状态。
- 禁止输入缓冲区。读这些禁止的缓冲区的任何位返回 0。
- 禁止上拉电阻。

9.5.3 硬件触发

ADC 模块有一个可选的异步硬件转换触发器， ADHWT，当 ADTRG 位置位时，它被允许。关于该 MCU 的 ADHWT 源的具体细节参考模块介绍。

当 ADHWT 源可用并且硬件触发被允许 （ADTRG=1），在 ADHWT 的上升沿初始化转换。如果当一个上升沿产生，一个转换正在处理中，上升沿被忽略。在连续转换设置中，只有引起连续转换的首次上升沿可以被发现。硬件触发功能和任何转换模式和设置协力完成操作。

9.5.4 转换控制

可以使用 8 位或 10 位模式转换，由 MODE 位决定。一个软件或硬件触发可以初始化转换。另外， ADC 模块可以设置为低功耗操作，长采样时间，连续采样，自动比较转换值和软件决定的比较值。

9.5.4.1 初始化转换

满足以下条件，即初始化转换：

- 如果选择软件触发操作，在写 ADCSC1 之后 （ADCH 不是全 1）。
- 如果选择硬件触发操作，在一个硬件触发 （ADHWT）事件之后。
- 当允许连续转换时，在将数据传到数据寄存器之后。

如果允许连续转换，当前转换完成后，一个新的转换可以自动初始化。在软件触发中，连续转换在写 ADCSC1 后开始，并继续直到终止。在硬件触发操作中，连续转换在硬件触发事件后开始，并继续直到终止。

9.5.4.2 完成转换

当转换的结果传到数据结果寄存器， ADCRH 和 ADCRL 后，转换完成。通过置位 COCO 表示。如果 AIEN 是高，在 COCO 置位时会产生一个中断。

在 10 位模式中，如果数据正在被读 （ADCRH 已经被读但是 ADCRL 还未被读），闭锁机制保护新的数据不会重写在 ADCRH 和 ADCRL 中以前的数据。当锁有效，数据传送被锁， COCO 不能置位，新的数据丢失。在允许比较功能的单个转换的情况下，并且比较条件为假，锁没有作用， ADC 操作被终止。在其他情况下，当数据传送被锁，除非 ADCO 的状态 （允许单个或连续转换），另一个转换被初始化。

如果允许单个转换，闭锁机制可能导致丢弃几个转换并且额外的功耗。为了避免这种情况，在初始化一个单个转换后，数据寄存器直到转换完成才能读。

9.5.4.3 终止转换

下列情况发生时，任何正在出来的转换都会终止：

- 写 ADCSC1 发生 （如果 ADCH 不是全 1，当前的转换被取消并开始了一个新的转换，）。
- 写 ADCSC2、ADCCFG、ADCCVH 或 ADCCVL 发生。这表明转换模式发生改变，因此当前的转换无效。
- MCU 复位。
- MCU 进入停止模式并且禁止 ADACK。

当一个转换终止，数据寄存器 （ADCRH 和 ADCRL）的内容不会改变，而是上次转换后完成后的传送的值。在因复位导致的转换终止情况中， ADCRH 和 ADCRL 返回到它们的复位值。

9.5.4.4 电源控制

直到初始化一个转换， ADC 模块都保持空闲。如果 ADACK 被选作转换时钟源， ADACK 时钟产生器也被允许。

当有效时功耗可以通过设置 ADLPC 减小。这导致更小的 f_{ADCK} 最大值 （参考电气描述）。

9.5.4.5 总转换时间

总转换时间依赖于抽样时间 （由 ADLSMP 决定）， MCU 总线频率，转换模式 （8 位或 10 位），转换时钟的频率 （f_{ADCD}）。模块有效后，输入的采样开始。 ADLSMP 用于选择长或短采样时间。当转换完成，转换器和输入通道隔离，用逐次渐进算法将模拟信号转换成数字值。转换算法完成后，转换结果传送到 ADCRH 和 ADCRL。

如果总线频率小于 f_{ADCK} 频率，当允许短采样 （ADLSMP=0）时，无法保证连续转换的精确采样时间。如果总线频率小于 f_{ADCK} 频率的 1/11，当允许长采样 （ADLSMP=1）时，无法保证连续转换的精确采样时间。

表 9-13 中总结了不同条件下的最大的总转换时间。

表 9-13. 不同控制条件件的总转换时间

转换类型	ADICLK	ADLSMP	最大总转换时间
单个或第一个连续转换 8 位	0x， 100	0	20ADCK 周期 +5 总线时钟周期
单个或第一个连续转换 10 位	0x， 100	0	23ADCK 周期 +5 总线时钟周期
单个或第一个连续转换 8 位	0x， 100	1	40ADCK 周期 +5 总线时钟周期
单个或第一个连续转换 10 位	0x， 100	1	40ADCK 周期 +5 总线时钟周期
单个或第一个连续转换 8 位	11	0	5s+20ADCK +5 总线时钟周期
单个或第一个连续转换 10 位	11	0	5s+23ADCK +5 总线时钟周期
单个或第一个连续转换 8 位	11	1	5s+40ADCK +5 总线时钟周期
单个或第一个连续转换 10 位	11	1	5s+43ADCK +5 总线时钟周期
后来的连续转换 8 位 $f_{BUS} \geqslant f_{ADCK}$	xx	0	17ADCK 周期
后来的连续转换 10 位 $f_{BUS} \geqslant f_{ADCK}$	xx	0	20ADCK 周期

表 9-13. 不同控制条件的总转换时间 （续）

后来的连续转换 8 位 $f_{BUS} \geqslant f_{ADCK}/11$	xx	1	37ADCK 周期
后来的连续转换 10 位 $f_{BUS} \geqslant f_{ADCK}/11$	xx	1	40ADCK 周期

最大的总转换时间由转换时钟和分频因子决定。时钟源由 ADICLK 位决定，分频因子由 ADIV 描述。例如，在 10 位模式中，选择总线时钟作为输入时钟源，输入时钟除以 1 分频， 8MHz 总线时钟，则单个转换的转换时间是：

$$\text{转换时间} = \frac{23\ \text{ADCK}\quad\text{周期}}{8\,\text{MHz}\quad/1} + \frac{5\,\text{总线周期}}{8\,\text{MHz}} = 3.5\,\mu\text{s}$$

公式 9-2

总线时钟的个数 =3.5 μs × 8 MHz = 28 （周期）

注意

ADCK 频率必须在 ADC 说明书中的 f_{ADCK} 最小值和 f_{ADCK} 最大值之间。

9.5.5 自动比较功能

比较功能可以设置为检测上限或下限。采样和转换输入后，结果和比较值 （ADCCVH 和 ADCCVL）的补数相加。比较上限时 （ACFGT = 1），如果结果大于或等于比较值， COCO 置位。比较下限时 （ACFGT=0），如果结果小于比较值， COCO 置位。转换结果和比较值的补数相加后产生的值传送到 ADCRH 和 ADCRL。

注意

当 MCU 在等待或 stop3 模式时，比较功能用于监控通道上的电压。满足比较条件时， ADC 中断将唤醒 MCU。

9.5.6 MCU 等待模式操作

WAIT 指令使 MCU 进入低功耗待命模式。因为时钟源仍然活动，这种模式可以很快恢复。如果 MCU 进入等待模式时，有一个转换正在处理，它将继续直到完成。当 MCU 处于等待模式时，通过硬件触发的方式或者如果允许连续转换，可以初始化转换。

处于等待模式时，总线时钟，总线时钟的一半和 ADACK 可以作为转换时钟源。在等待模式时， ALTCLK 作为转换时钟源使用是由该 MCU 的 ALTCLK 的定义决定的。参考该 MCU 中模块说明中关于 ALTCLK 说明的信息。

9.5.7 MCU stop3 模式操作

在 MCU 中禁止了大多数或所有的时钟源期间， STOP 指令可以使 MCU 进入低功耗待命模式。

9.5.7.1 禁止 ADACK 的 stop3 模式

如果不选择异步时钟 ADACK 作为转换时钟，执行 STOP 指令终止当前转换并且使 ADC 进入空闲状态。stop3 模式不影响 ADCRH 和 ADCRL 的值。从 stop3 模式退出后，需要一个软件或硬件触发重新开始转换。

9.5.7.2 允许 ADACK 的 stop3 模式

如果选择 ADACK 作为转换时钟，在 stop3 模式时， ADC 继续工作。为了保证 ADC 操作， MCU 的电源调整器在 stop3 模式时必须仍然有效。参考该 MCU 模块介绍中的配置信息。

如果 MCU 进入 stop3 模式时，有一个转换正在处理，它将继续直到完成。当 MCU 处于 stop3 模式时，通过硬件触发的方式或者如果允许连续转换，可以初始化转换。

转换完成事件置位 COCO，并且如果中断允许 （AIEN = 1）还会产生一个 ADC 中断将 MCU 从等待模式唤醒。

注意

ADC 可能将系统从低功率停止中唤醒，导致 MCU 开始强烈的运行电平电流而没有产生一个系统电平中断。为了避免这种情况，当进入 stop3 模式并继续 ADC 转换时，软件应该确保数据传输闭锁机制 （在第 9.5.4.2 节 完成转换）清零。

9.5.8 MCU stop1 和 stop2 模式操作

当 MCU 进入 stop1 或 stop2 模式时，自动禁止 ADC 模块。从 stop1 或 stop2 退出时，所有的模块寄存器存放的是复位值。因此从 stop1 或 stop2 退出时，模块必须被重新使能和重新配置。

9.6 初始化信息

该节举例说明了用户如何初始化和配置 ADC 模块的一些基本方用法。用户可以灵活地选择配置模块， 8 位或 10 位精度，单个或连续转换，循环或中断方式，还有其他选项。在该例中的信息可以参见表 9-7、表 9-8 和表 9-9。

注意

十六进制的前缀是 0x，二进制的前缀是 %，十进制没有前缀。

9.6.1 ADC 模块初始化举例

9.6.1.1 初始化顺序

在 ADC 模块执行转换操作前，必须初始化。典型的初始化顺序是：

1. 更新配置寄存器 （ADCCFG）选择输入时钟源和产生内部时钟 （ADCK）的分频因子。这个寄存器也可以用来选择采样时间和低功耗配置。
2. 更新状态和控制寄存器 2 （ADCSC2）选择转换触发器 （硬件或软件）和比较功能选项 （如果允许）。
3. 更新状态和控制寄存器 1 （ADCSC1）可以选择转换是否是连续的还是仅一次完成，并且允许或禁止转换完成中断。选择哪路输入通道完成转换也是在这里操作。

9.6.1.2 伪代码举例

在该例中， ADC 模块将提供允许中断产生一个单个 10 位转换，低功耗，输入通道 1 上的长采样时间，内部的 ADCK 时钟将由总线时钟除以 1 得到

ADCCFG=0x98 （%10011000）

Bit 7	ADLPC	1	配置为低功耗 （降低最大的时钟速率）
Bit 6:5	ADIV	00	设置 ADCK 为输入时钟 /1

Bit 4	ADLSMP	1	设置位长采样时间
Bit 3:2	MODE	10	设置位 10 位转换模式
Bit 1:0	ADICLK	00	选择总线时钟作为输入时钟

ADCSC2=0x00 （%00000000）

Bit 7	ADACT	0	标志表示转换是否在处理中
Bit 6	ADTRG	0	选择软件触发器
Bit 5	ACFE	0	禁止比较功能
Bit 4	ACFGT	0	在该例中未使用
Bit 3:2		00	未实现或保留，读为 0
Bit 1:0		00	保留为 Freescale 使用；写为 0

ADCSC1=0x41 （%01000001）

Bit 7	COCO	0	只读标志当转换完成时置位
Bit 6	AIEN	1	转换完成中断使能
Bit 5	ADCO	0	仅一次转换 （禁止连续转换）
Bit 4:0	ADCH	00001	选择通道 1 作为 ADC 输入通道

ADCRH/L = 0xxx

保存转换结果。在低字节前读高字节，所以转换数据不会被下一次转换的数据重写。

ADCCVH/L = 0xxx

当允许比较功能时，保存比较值

APCTL1=0x02

禁止 AD1 引脚 I/O 控制。其他 AD 引脚仍然位通用 I/O 引脚

APCTL2=0x00

其他 AD 引脚仍然时通用 I/O 引脚。

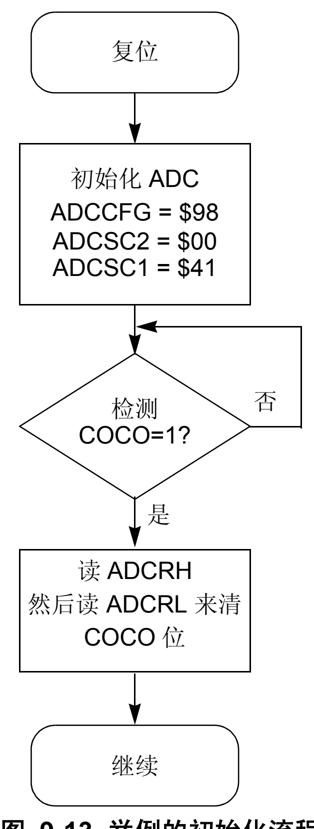


图 9-13. 举例的初始化流程

9.7 应用信息

该节包含了在应用中使用 ADC 模块的信息。 ADC 被设计集成了一个微控制器可以使用在需要 A/D 转换器的嵌入式控制应用中。

9.7.1 外部引脚和安排

下面讨论了和 ADC 模块相关的外部引脚和如何最好的使用它们。

9.7.1.1 模拟电源引脚

ADC 模块有电源和地引脚 （V_{DDAD} 和 V_{SSAD}），在一些设备中有独立的引脚。在其他设备， V_{SSAD} 和 MCU V_{SS} 复用相同的引脚。在一些设备中， V_{SSAD} 和 V_{DDAD} 共享数字供电引脚。在这些情况中，有独立模拟供电，和相应的数字电源绑定在相同的引脚。所以这两个电源在一定程度上保持隔离。

当作为独立的引脚时， V_{DDAD} 和 V_{SSAD} 必须和相应的 MCU 数字电源 （V_{DD} 和 V_{SS}）连接相同的电压，谨慎布线避免干扰，旁路电容离封装尽可能近。

在模拟和数字电源单独供电时的情况，这两个电源的接地连接必须是 V_{SSAD} 引脚。如果可能这应该是这两个电源的唯一接地连接。 V_{SSAD} 接单个的地位置。

9.7.1.2 模拟参考引脚

除了模拟电源， ADC 模块连接了两个参考电压输入。高参考电源是 V_{REFH}，在有些设备中，可能和 V_{DDAD} 复用相同的引脚。地参考电压是 V_{REFL}，在有些设备中和 V_{SSAD} 复用相同的引脚。

当作为单独引脚时， V_{REFH} 可能和 V_{DDAD} 连到相同的电压。或可能是由外部得到 （介于 V_{DDAD} 最小值和 V_{DDAD} 之间， V_{REFH} 绝对不能大于 V_{DDAD}）。当作为单独引脚时， V_{REFL} 可能和 V_{DDAD} 连到相同的电压。V_{REFH} 和 V_{REFL} 必须谨慎布线防止最大干扰度和旁路电容离封装尽可能近。

在每次逐次渐进步骤，电流尖峰组成的交流 （AC）电流通过 V_{REFH} 和 V_{REFL} 循环位电容阵列提供电荷。满足这个电流要求的最好的外部元件是 0.1μF 的电容 （高频特性）。这个电容连接到 V_{REFH} 和 V_{REFL} 之间，离封装尽可能近。不推荐使用电阻，因为电流导致电压泄露，这可能导致转换错误。该路径上的电磁感应应最小化。

9.7.1.3 模拟输入引脚

外部模拟输入通常和 MCU 设备的 I/O 引脚复用。通过置位引脚控制寄存器的相应位可以禁止引脚 I/O 控制。相应的引脚控制寄存器位没有置位，可以进行转换操作。当引脚作为模拟输入时，推荐置位引脚控制寄存器位。这避免了连接问题，因为输入缓冲区处于高阻抗状态并且禁止上拉电阻。而且，当输入既不是 V_{DD} 也不是 V_{SS}，输入缓冲吸收直流 （DC）电流。置位引脚控制位，所有的引脚作为模拟输入，可以达到最低的操作电路。

试验数据表明，当存在噪音或源阻抗高时，模拟输入上的电容可以提高性能。试验 0.01 μ F 的电容 （有高频特性）完全可以满足。这些电容并不是在所有的情况下都需要，但是它们必须放在离封装尽可能近的地方，作为 V_{SSA} 的参考。

为正确转换，输入电压必须在 V_{REFH} 和 V_{REFL} 之间。如果输入等于或大于 V_{REFH} 转换电路把信号转换成 $3FF （共 10 位表示）或 $FF （共 8 位表示），如果输入等于或小于 V_{REFH}，转换电路把信号转换成 $000，在 V_{REFH} 和 V_{REFL} 之间的输入电压是线性转换。当采样电容正在充电时，将会有一个和 V_{REFL} 相关的短暂电流。当 ADLSMP 低，将会采样以 ADCK 为源的 3.5 个周期，当 ADLSMP 高，则是 23.5 个周期。

为了减少因电流进入而引起的正确度减小，连接到模拟输入的引脚在转换期间不应该传输。

9.7.2 错误源

A/D 转换中存在几种错误源。它们在该节的后面讨论。

9.7.2.1 采样错误

为正确转换，输入必须被采样足够长时间才能达到合适的精度。如果最大输入电阻 7 kΩ 和输入电容 5.5 pF，外部模拟源 （RAS）的电阻小于 5 kΩ，则采样 1/4LSB （10 位精度）可以在最小的采样窗口完成 （3.5 周期， 8MHz 最大 ADCK 频率）。

更高电阻或更高精度的采样可以通过置位 ADLSM （增加采样窗口到 23.5 个周期）或通过减少 ADCK 频率来增加采样时间。

9.7.2.2 引脚漏电流误差

如果外部模拟源 （RAS）为高， I/O 引脚上的漏电流导致转换误差。如果在应用中不能容忍这个错误，保持 RAS 小于 V_{DDAD} /(2N*I_{LEAK}) 将会后更少漏电流误差 （8 位模式 N = 8， 10 位模式 N = 10）

9.7.2.3 噪音误差

在采样或转换过程中产生的系统噪音会影响转换的正确性。只要满足下面指定的条件才能保证 ADC 采样的正确性。

- V_{REFH} 和 V_{REFL} 之间有一个 0.1 µF 的低内阴电容。
- V_{DDAD} 和 V_{SSAD} 之间有一个 0.1 µF 的低内阴电容。
- 如果电源使用感应隔离，一个 1 µF 的电容放在 V_{DDAD} 和 V_{SSAD} 之间。
- V_{SSAD} （和 V_{REFL}，如果连接）连接到 V_{SS} （连着地平面的点）。
- 初始化 （硬件触发转换）前或刚初始化 （软件或硬件触发转换） ADC 转换， MCU 处于等待或 stop3 状态。
  - 对于软件触发转换，用 WAIT 或 STOP 指令写 ADCSC1 后。
  - 对于 stop3 模式操作，选择 ADACK 作为时钟源。在 stop3 模式的操作减少 V_{DD} 噪音但是因为停止复原增加了有效转换时间。
- MCU 处于转换时，没有 I/O 选择，输入或输出。

在一些情况，外部系统行为导致辐射或噪音发射或伴随 ADC 的过多 V_{DD} 噪音。在这些情况，或当 MCU 不能在等待状态或 I/O 行为停止，上述操作可能减少影响正确性的噪音：

- 在选择的输入通道和 V_{REFH} 或 V_{REFL} 之间放置一个 0.01 µF 的电容 （CAS）（这将增加噪音问题但是影响基于外部的模拟源电阻的采样率）。
- 求多次转换模拟值的平均。需要四次采样减少 1LSB 的影响以及一次误差。
- 通过关闭异步时钟 （ADACK）和求平均，减少同步噪音的影响。和 ADCK 同步的噪音无法达到平均数。

9.7.2.4 编码宽度和量化误差

ADC 可以将输入的线性值量化成 1024 个块 （在 10 位模式）。每块有相同的高度 （1 个代码）和宽度。宽度定义为 dleta，在一个代码到下一个之间。 N 位转换器 （N 是 8 或 10）的理想代码宽度，定义为 1LSB，为：

$$1LSB=（V_{REFH} - V_{REFL}） /2^N \qquad \text{公式 9-3}$$

数字化结果存在固有的量化误差。对于 8 位或 10 位转换。用两点的中点表示电压，代码被转化。因此在 8 位或 10 位模式中，量化误差 ±1/2LSB。因此，第一个转化的编码宽度 （$000）只有 1/2LSB 并且最后一个转化的宽度时 1.5LSB。

9.7.2.5 线性误差

ADC 可能存在几种非线性的情况。各种方法减少这些错误，但是系统仍然会存在，因为它们影响全局的精度。这些错误是：

- 归零误差 （EZS）（也称作偏移量）——指第一个转换的实际编码宽度和理想编码宽度的不同 （1/2LSB）。注意如果第一个转换是 $001，则采用了然后实际的 $001 编码宽度和理想情况 （1LSB）的不同。
- 满标误差 （EFS）——指最后一个转换的实际编码宽度和理想编码宽度的不同 （1.5LSB）。注意如果最后一个转换是 $3FE，采用了实际的 $001 编码宽度和理想情况 （1LSB）的不同。
- 微分非线性 （DNL）——指所有转换中实际编码宽度和理想编码宽度最大误差。

- 积分非线性 （INL）—— 指 DNL 总和所能达到的最大值。更简单的，所有编码中，编码所给的实际的转换电压和和相应的理想电压的最大误差。
- 总非校准误差 （TUE）——这个错误定义为实际转换函数和理想线性转换函数的不同，因此包含所有形式的错误。

9.7.2.6 编码抖动、非单调性和遗编码

ADC 易受三种特殊形式的错误影响。它们是代码抖动，非单调性，遗编码。

代码抖动，在某一点时，当重复采样时，一个确定的输入电压转化成两个值中的一个。理想的，当输入电压比转换电压只小很少时，转换器产生更低的编码 （反之亦然）。然而，即使很小的系统噪音也会导致转换器对于转换电压周围一定范围的输入电压不确定 （在两个编码之间）。这个范围通常是 1/2LSB，并且随噪音而增加。这个误差可以通过重复采样和对结果求均值来减少。另外在 9.7.2.3 节 噪音误差节中的技巧可以减少该误差。

非单调性可以定义为，除了代码抖动，转换器可能转换一个较高电压位较低编码。遗编码是那些对于任何输入都不会转换的值。

在 8 位或 10 位模式中， ADC 将确保单调并且不遗失编码。

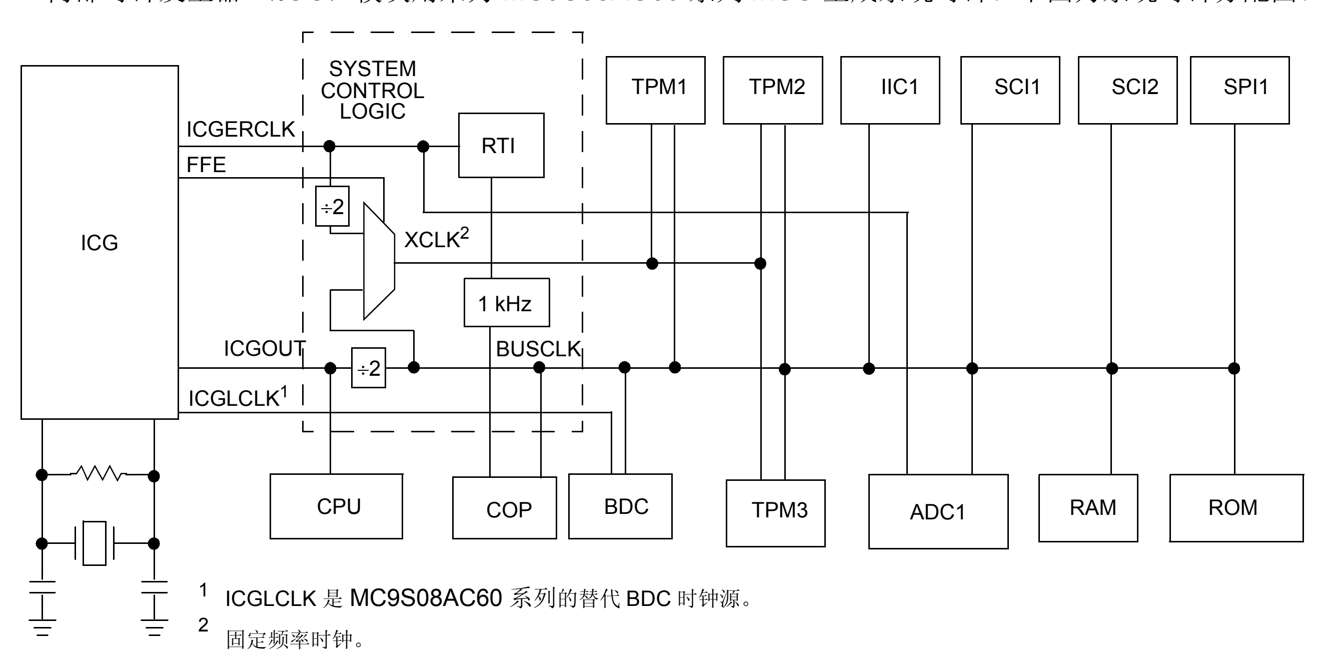
第 10 章
内部时钟发生器 (S08ICGV4)

10.1 介绍

ICG 提供多种时钟源，这使用户能够非常灵活地根据成本、精确度、电流消耗及性能要求进行选择。如 图 10-2 所示，ICG 包含四个功能子模块块。下面对每个子模块进行了简要的描述，更详细的描述参见后面的章节。.

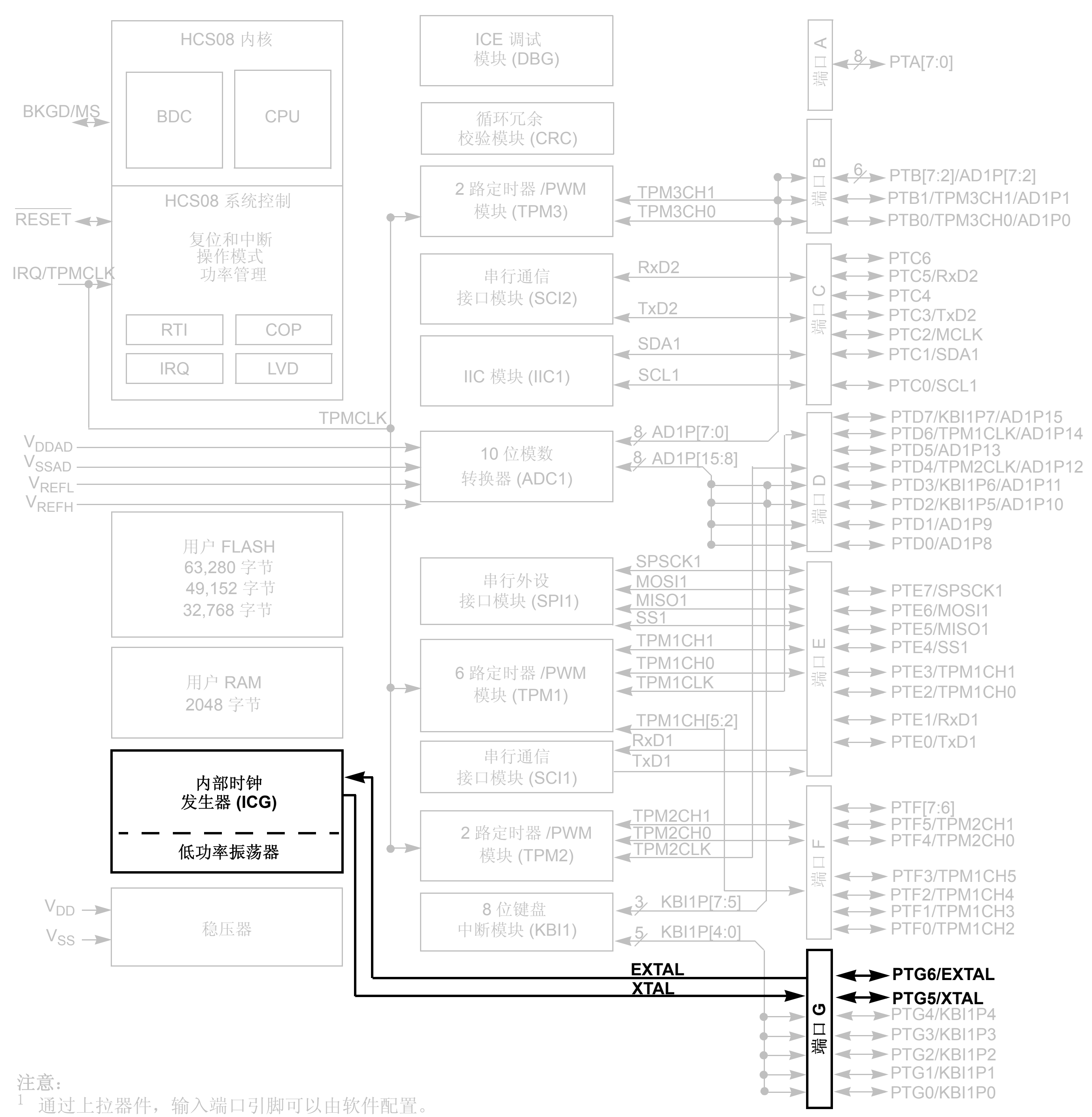
- **振荡器子模块** — 振荡器子模块提供了连接外部晶体或谐振器的手段。两个频率范围可以通过软件选择，实现最优的启动和稳定性。另外，振荡器子模块还可以用来向系统时钟发送外部方波。外部源可以提供非常精确的时钟源。振荡器能够通过配置 HGO 用于低功率模式或高振幅模式。
- **内部参考发生器** — 内部参考发生器包含两个受控制的时钟源。一个约为 8MHz，可以作为后台调试控制器的内部时钟。其它内部参考时钟源一般为 243 kHz ，当精确定时的事件输入到 MCU 时，可以通过软件进行修整，提高精确度。这提供了高可靠性，低成本的时钟源。
- **锁频环** — 锁频环 （FLL）子模块可采用内部或外部时钟源，并可以倍增它至更高的频率。状态位提供电路锁定和失锁状态的信息。此外，这个子模块可以监控外部参考时钟和信号，判断该时钟是否为有效时钟。
- **时钟选择功能子模块** — 时钟选择功能子模块提供多种开关选择，用于将不同的时钟源连接到系统时钟树。ICGDCLK 是在 FLL 基础上倍增的时钟频率， ICGERCLK 是来自晶振或外部时钟源的参考时钟频率， FFE （固定时钟频率使能）是一个控制信号，用来控制系统固定频率时钟 (XCLK)。 ICGLCLK 是后台调试控制器 （BDC）的时钟源。

内部时钟发生器 （ICG）模块用来为 MC9S08AC60 系列 MCU 生成系统时钟。下图为系统时钟分配图。



1 ICGLCLK 是 MC9S08AC60 系列的替代 BDC 时钟源。
2 固定频率时钟。

图 10-1. 系统时钟分配图



注意：

[1] 通过上拉器件，输入端口引脚可以由软件配置。

[2] 如果 IRQ 使能 (IRQPE = 1)，引脚包含软件可配置的上拉 / 下拉器件。如果选择上升边沿检测，则下拉使能 (IRQEDG = 1)。

[3] 引脚包含集成上拉器件。

[4] PTD3、PTD2、PTD7 和 PTG4 既包含上拉也包含下拉器件。当 KBI 使能时 (KBIPEn = 1)，且选择了上升边沿 (KBEDGn = 1)，则下拉使能 。

[5] TPMCLK、 TPM1CLK 和 TPM2CLK 选项可通过软件配置； 复位后， TPM1CLK、 TPM2CLK 和 TPMCLK 分别提供给 TPM1、 TPM2 和 TPM3。

图 10-2. 显亮 ICG 模块及其引脚的 MC9S08AC60 系列模块图

10.2 概述

ICG 提供多个时钟源。这为用户在均衡成本、精度、电流消耗量以及性能时，提供了极大的灵活性，如图 10-3 所示， ICG 由四个功能模块组成。下面将对这些模块进行简要的描述，详细描述请参见后续章节。

- 振荡器模块 — 振荡器模块用于连接外部晶振或振荡器。它提供两种可软件选择的频率范围，以获得最佳的启动和稳定性。振荡器模块可以产生一个外部方波给系统时钟。外部时钟源可以提供非常精确的时钟。 HGO 可以配置振荡器为低功耗模式或高幅模式。
- 内部辅助发生器 —内部辅助发生器由两个受控时钟源组成。一个为大约 8 MHz 作为背景调试控制器的本地时钟。另一个时钟源是典型的 243 kHz，当一个精确的定时事件输入到 MCU 时，它可通过软件进行微调 (Trim) 以提高其精确度。这将提供一个非常可靠，低成本的时钟源。
- 锁频环 — 对内部或外部时钟源进行倍频的时候，需要用到锁频环。当电路频率锁定或失锁时，状态位将提供标志信息。另外这个模块可以监控外部参考时钟及时钟信号是否有效。
- 时钟选择模块 —时钟选择模块为不同时钟源连接到系统时钟树提供几种开关选择。 ICGDCLK 是 FLL 的输出倍频时钟， ICGERCLK 是源自晶振或外部时钟源的参考时钟频率， FFE(固定频率使能) 是一个用来控制系统固定频率时钟 （XCLK）的控制信号。 ICGLCLK 是背景调试控制器的时钟源。

10.2.1 特性

该模块非常易用，许多功能无需用户干预自动实现。若需要快速配置此模块，请参考 10.6 节 初始化 / 应用信息，并选取一个与应用相似的例子。

ICG 和时钟分配系统的特性如下：

- 几种可选主时钟源提供灵活的成本，频率和精确度选择。
  - 32 kHz ～ 100 kHz 的晶振或振荡器
  - 1 MHz ～ 16 MHz 的晶振或振荡器
  - 外部时钟
  - 内部参考发生器
- 缺省设置为自时钟模式以获得最小的启动延时
- 频率锁定回路 （FLL）产生 8 MHz ～ 40 MHz 时钟 (总线时钟最高至 20 MHz)—用外部或内部时钟作参考频率。
- 自动停止非工作状态的时钟源。
- 丢失时钟或者 FLL 失锁会产生复位或中断
- 数字控制振荡器 （DCO）保留先前的频率设定，当从 stop3 模式恢复时允许快速频率锁定。
- 在丢失或移除参考时钟时， DCO 将维持操作频率。
- 后 FLL 分频器选择 8 个总线分频约数 (/1 到 /128) 中之一。
- 实时中断的独立自时钟源。
- 可调整的内部时钟源支持串行通讯，无需额外的外部部件。
- 锁获得后自动 FLL 接触。
- 外部振荡器可选择低功率或高增益

10.2.2 操作模式

本节对操作模式进行简单描述。详细描述参见 10.5 节 功能描述。

- 模式 1 — OFF

输出时钟， ICGOUT 是静态的，当 STOP 指令被执行时，将进入此模式。

- 模式 2 — 自时钟 （SCM）

  模式 2 是缺省模式，在复位后立即进入此模式。内部时钟发生器的 FLL 是开环的，数字控制振荡器自由运行在设置的频率。

- 模式 3 — FLL 参与模式 (FEI)

  在此模式下， FLL 对内部时钟源进行可编程倍频来产生频率。

  - 当 FLL 试图上锁时，未上锁的 FEI 是一个暂态。此时， FLL 的数字控制振荡器 (DCO) 频率偏离目标频率 , 而 FLL 调整数字控制振荡器以达到目标频率。
  - 当 FLL 检测到数字控制振荡器锁定为目标倍频频率时 ,FLL 将上锁。

- 模式 4—FLL 外部旁路

  在这种模式下， ICG 被配置为旁路 FLL，而使用外部时钟作为系统时钟源。

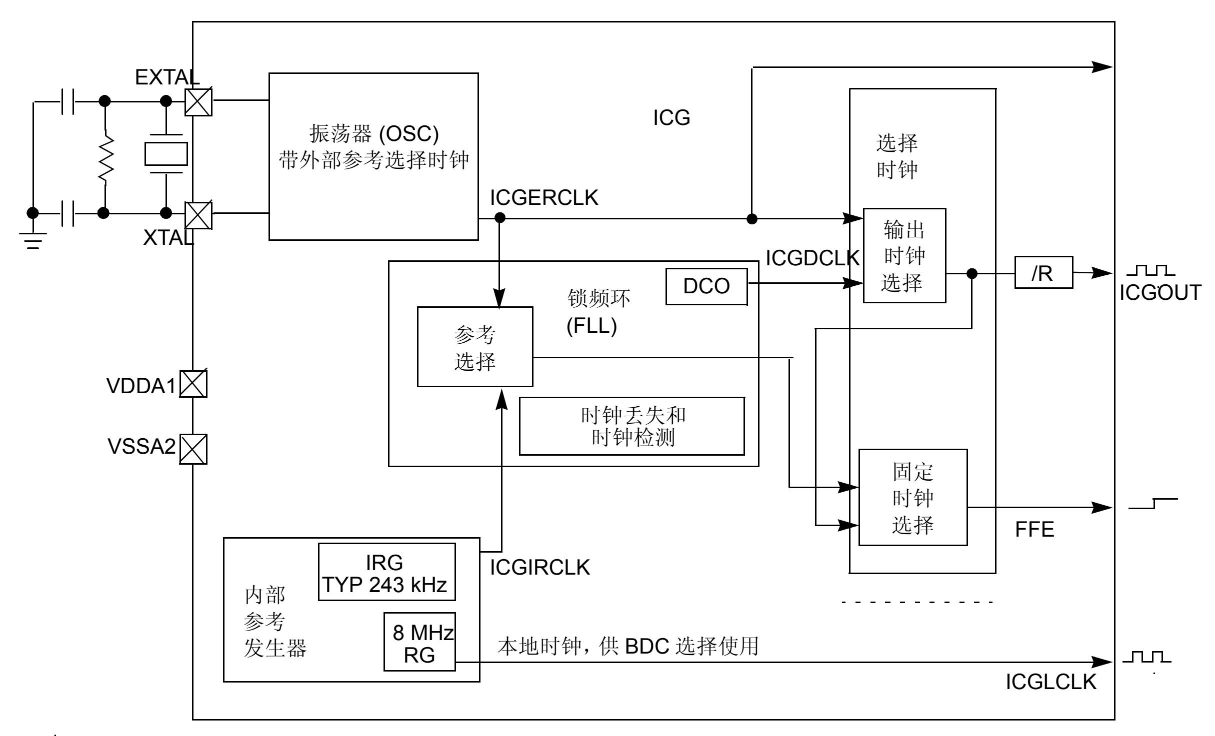
- 模式 5—外部使用的 FLL （FEE）

  内部时钟发生器的 FLL 对外部时钟源进行倍频

  - 当 FLL 试图上锁时，未上锁的 FEI 是一个暂态。此时， FLL 的数字控制振荡器 (DCO) 频率偏离目标频率 , 而 FLL 调整数字控制振荡器以达到目标频率。
  - 当 FLL 检测到数字控制振荡器锁定为目标倍频频率时， FLL 将上锁。

10.2.3 功能结构图

图 10-3 是内部时钟发生器 （ICG）模块的功能结构图。



1 不是所有的 HCS08 芯片都含有 ICG 单独的电源引脚。参阅设备引脚分配。

图 10-3. ICG 框图

10.3 外部信号描述

该振荡器引脚用来提供一个外部时钟源给 MCU。振荡器引脚在低功耗模式下（缺省）其增益受控。在低功耗模式下振荡器振幅限制为约 1 V（峰值与峰值之间）。

10.3.1 EXTAL— 外部参考时钟 / 振荡器输入

若是第一次写 ICGC1，无论选择 FEE 模式或者 FBE 模式，该引脚将由 REFS 决定作为外部时钟输入或振荡器电路输入。若是第一次写 ICGC1，无论选择 FEI 模式或者 SCM 模式，该引脚都不被 ICG 使用。

10.3.2 XTAL—振荡器输出

若是第一次写 ICGC1，无论选择 FEE 模式或者 FBE 模式，该引脚将被作为振荡器电路输出。若是第一次写 ICGC1，无论选择 FEI 模式或者 SCM 模式，该引脚将不被 ICG 使用。振荡器能够被配置提供更高振幅输出，降低噪音影响。这种操作模式通过置 HGO = 1 来选择。

10.3.3 外部时钟连接

若使用外部时钟，则引脚连接如图 10-4 所示。

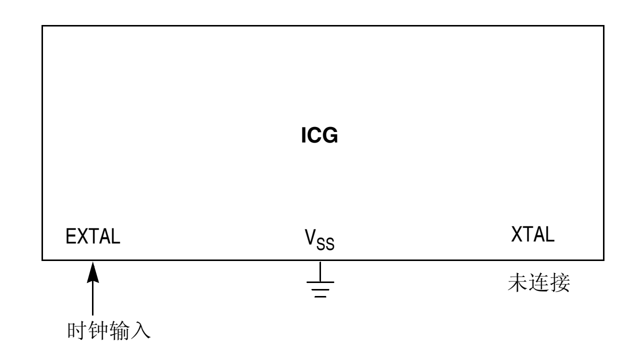


图 10-4. 外部时钟连接

10.3.4 外部晶振 / 谐振器连接

如果使用外部晶振 / 谐振器，那么引脚可按图 10-5 连接。推荐的元件值被列在附录 A 电气特性和时序规范。

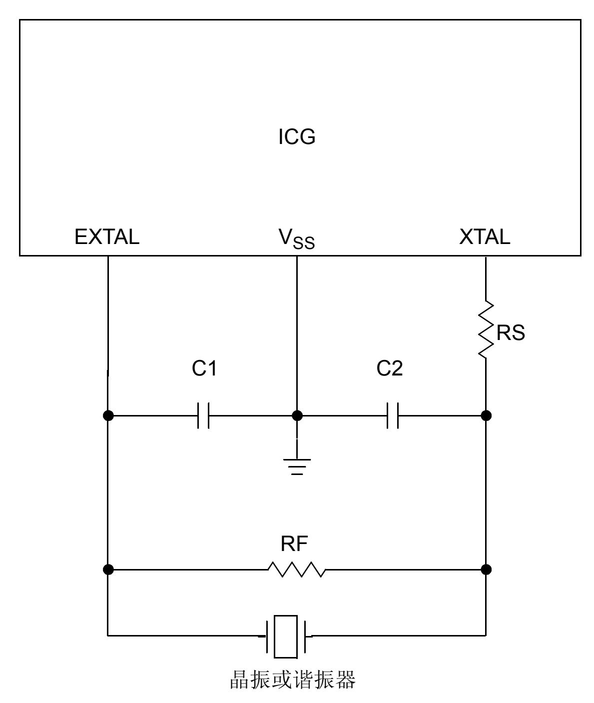
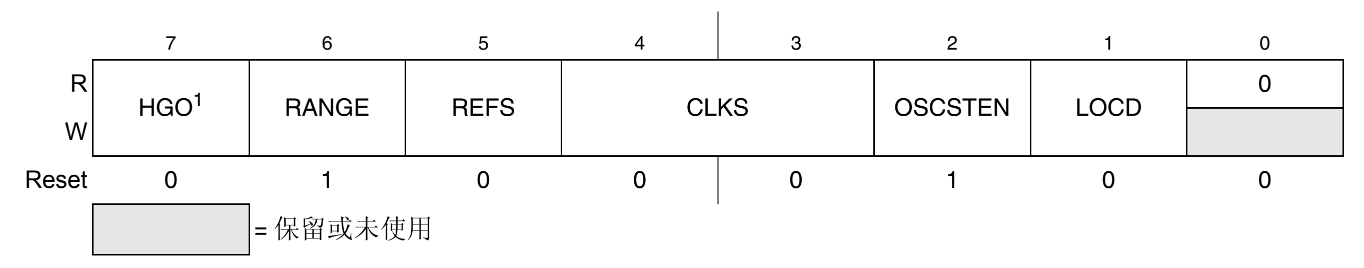


图 10-5. 使用外部参考频率的连接

10.4 寄存器定义

参阅存储器节的直接页寄存器，可以找到所有 ICG 寄存器绝对地址分配表。这部分通过名称检索寄存器和控制位。飞思卡尔提供一个头文件用来把名称转换为相应的绝对地址。

10.4.1 ICG 控制寄存器 1 （ICGC1）



	7	6	5	4	3	2	1	0
R	HGO[1]	RANGE	REFS	CLKS		OSCSTEN	LOCD	0
W								
Reset	0	1	0	0	0	1	0	0

= 保留或未使用

1 该位在复位后仅可写 1 次。之后的写操作被忽略。

图 10-6. ICG 控制寄存器 1 （ICGC1）

表 10-1. ICGC1 寄存器位描述

位	描述
7 HGO	高增益振荡器的选择位—该位用来选择低功耗操作或高增益操作以降低噪声影响。该位复位后只可写一次。 0 振荡器配置为低功耗操作 1 振荡器配置为高增益操作
6 RANGE	频率范围选择位—该位控制振荡器，参考分频器， FLL 环路分频器倍频因子 （P）。它为 ICG 选择频率范围。该位复位后只可写一次。该位仅在外部使用 FLL(FEE) 和 FLL 外部旁路 (FBE) 模式有效。 0 振荡器配置为低频率范围。 P = 64 1 振荡器配置为高频率范围。 P = 1
5 REFS	外部参考源选择—该位控制 ICGERCLK 的外部参考时钟源。该位复位后只可写一次。 0 外部时钟请求 1 振荡器使用晶振或振荡器请求
4:3 CLKS	时钟模式选择—该位控制时钟模式。如果请求 FLL 外部旁路模式 (FBE)，除非 ERCS=1，否则该位将不能被选择。如果 ICG 关闭，该位将保持不变。如果先前写未完成，则写 CLKS 将无效。 00 自时钟 01 内部参考使用的 FLL 10 外部参考旁路 FLL 11 外部参考使用的 FLL CLKS 位在任何时间都可写，除非在复位后的第一次写为 CLKS = 0X， CLKS 位不能被写为 1X 直到下个复位操作之后。
2 OSCSTEN	在 OFF 模式下振荡器使能位—当 ICG 关闭时，该位控制振荡器电路是否保持使能。如果 HGO = 1 并且 RANGE = 1，该位无效。 0 当 ICG 处在 OFF 模式时，振荡器禁止，除非使能位为高， CLKS = 10 并且 REFST = 1。 1 当 ICG 处于 OFF 模式时，振荡器被使能， CLKS = 1X 并且 REFST = 1。
1 LOCD	禁用时钟丢失检测 0 时钟丢失检测允许 1 时钟丢失检测禁止

10.4.2 ICG 控制寄存器 2 （ICGC2）

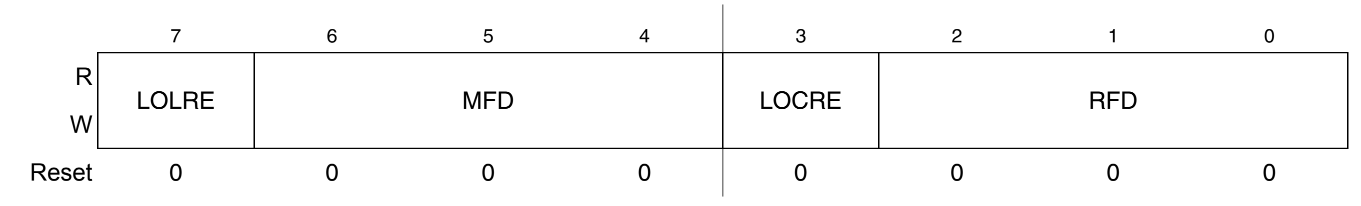


图 10-7. ICG 控制寄存器 2 （ICGC2）

表 10-2. ICGC2 寄存器位描述

位	描述
7 LOLRE	失锁复位使能位—该位决定当失锁发生的时候， ICG 发送的请求。只有当 LOLS 被置位时，该位才有效。 0 失锁产生一个中断请求 1 失锁产生一个复位请求
6:4 MFD	倍频因子—该位控制环路 FLL 中可编程倍频因子。由 MFD 位指定的值确定应用于参考频率的倍频因子 （N）。如果先前的一个写操作未完成，则写 MFD 位不产生影响。为 N 选择一个充分低的值以确保 $f_{ICGDCLK}$ 不会超出它的最大值。 000 倍频因子 =4 001 倍频因子 =6 010 倍频因子 =8 011 倍频因子 =10 100 倍频因子 =12 101 倍频因子 =14 110 倍频因子 =16 111 倍频因子 =18
3 LOCRE	时钟丢失复位使能—该位决定系统如何管理一个时钟丢失。 0 时钟丢失产生一个中断请求 1 时钟丢失产生一个复位请求
2:0 RFD	分频器—该位在时钟选择电路后控制分频器值。由 RFD 位指定的值确定分频因子用于选定的输出时钟源。如果先前的写操作未完成，则写 RFD 位不产生影响。 000 分频因子 =1 001 分频因子 =2 010 分频因子 =4 011 分频因子 =8 100 分频因子 =16 101 分频因子 =32 110 分频因子 =64 111 分频因子 =128

10.4.3 ICG 状态寄存器 1 （ICGS1）

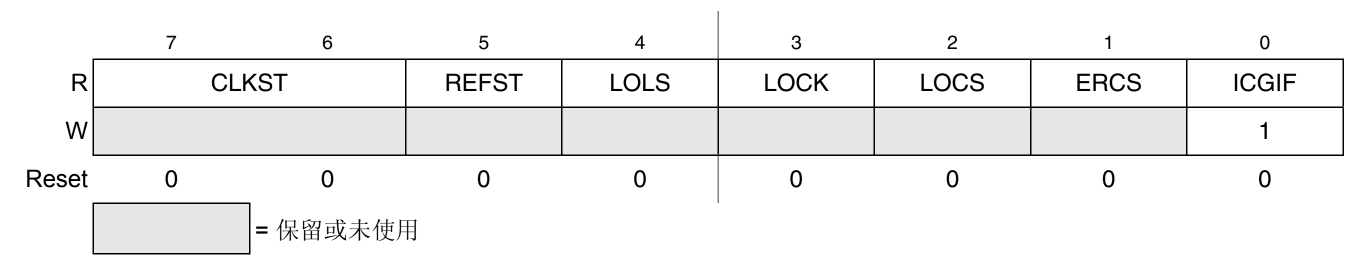
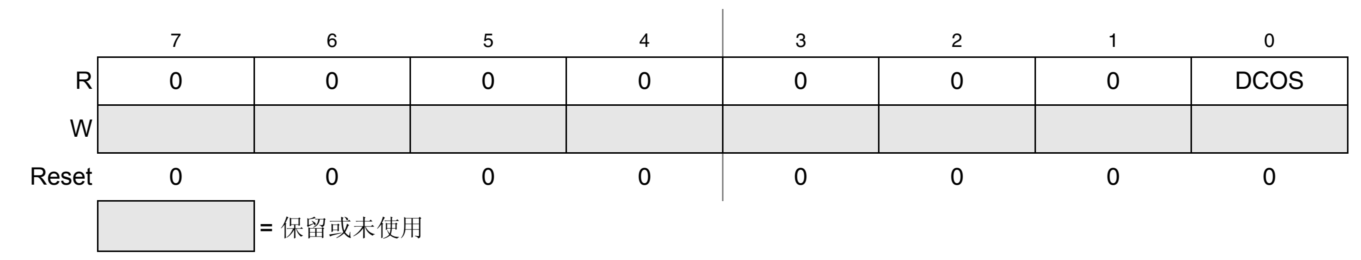


图 10-8. ICG 状态寄存器 1 （ICGS1）

表 10-3. ICGS1 寄存器位描述

位	描述
7:6 CLKST	时钟模式状态位—该位标识当前时钟模式。在写 CLKS 位后由于需要时钟域之间的内部同步， CLKS 位不会立即更新。 00 自时钟 01 内部参考使用的 FLL 10 外部参考旁路 FLL 11 外部参考使用的 FLL
5 REFST	参考时钟状态—该位标识时钟参考选择电路当前使用的时钟参考源。 0 选择外部时钟 1 选择晶振 / 振荡器
4 LOLS	FLL 失锁状态—该位是 FLL 锁状态的关联指示。 0 自从 LOCS 最近一次被清零后， FLL 没有意外丢失锁。 1 自从 LOCS 最近一次被清零后， FLL 出现意外失锁， LOLRE 决定将采取的行动。
3 LOCK	FLL 锁状态—该位标识 FLL 是否已获锁定。在关闭、自时钟、 FLL 旁路模式下该位被清零。 0 FLL 当前未锁 1 FLL 当前被锁
2 LOCS	时钟状态丢失—该位标识 ICG 时钟状态丢失。 0 自从 LOCS 最近一次被清零后， ICG 没有丢失时钟。 1 自从 LOCS 最后一次被清零后， ICG 出现时钟丢失， LOCRE 决定后续采取的行动。
1 ERCS	外部参考时钟状态—该位标识外部参考时钟 （ICGERCLK）是否满足最小频率要求。 0 外部参考时钟不稳定，频率要求未满足 1 外部参考时钟稳定，频率要求满足
0 ICGIF	ICG 中断标识—当出现挂起的 ICG 中断请求时， ICGIF 读 / 写标识被置位。当 ICGIF 被置位时可通过复位或读取 ICG 状态寄存器，并向 ICGIF 位写 1 来清零。如果在清零序列完成之前另一个 ICG 中断发生，该序列被复位，因此在清零上一个中断的序列后， ICGIF 还将保持原设置。写逻辑 0 给 ICGIF 无影响。 0 无挂起的 ICG 中断请求 1 有一个 ICG 中断请求挂起

10.4.4 ICG 状态寄存器 2 （ICGS2）



	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	DCOS
W								
Reset	0	0	0	0	0	0	0	0

= 保留或未使用

图 10-9. ICG 状态寄存器 2 （ICGS2）

表 10-4. ICGS2 寄存器位描述

位	描述
0 DCOS	数字控制振荡器时钟稳定—当 DCO 时钟 (ICG2DCLK) 稳定时 DCOS 位被置位，表示计数错误没被超过两次连续采样的 nunlock 所改变，并且该 DCO 时钟不是静止的。当退出 OFF 模式，如果 CLKS=X1，该位用来决定何时转换到请求的时钟模式。它同时也用在自时钟模式来决定何时开始监视 DCO 时钟。进入 OFF 模式该位被清零。 0 DCO 时钟不稳定 1 DCO 时钟稳定

10.4.5 ICG 滤波器寄存器 （ICGFLTU、 ICGFLTL）

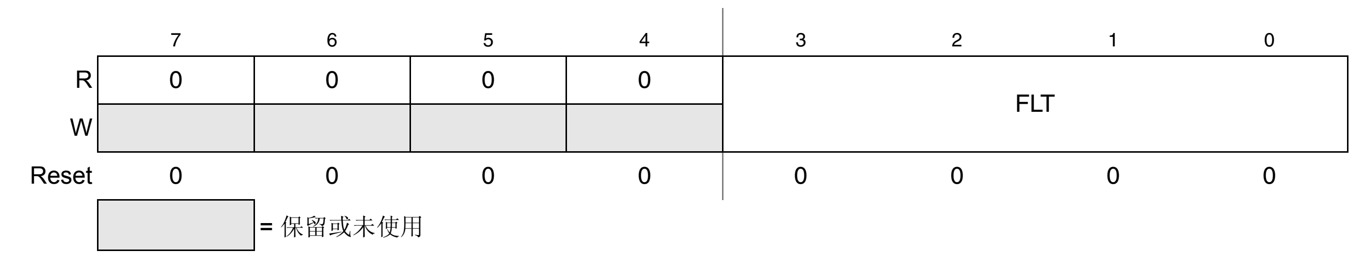


图 10-10. ICG 滤波寄存器高八位 （ICGFLTU）

表 10-5. ICGFLTU 寄存器位描述

位	描述
3:0 FLT	滤波器值—表明当前滤波器值，该值控制 DCO 频率。除了自时钟模式 （CLKS = 00），该位可读。在自时钟模式下任何写 ICGFLTU 将更新当前 12 位滤波器值。如果前一个关锁序列未完成，写 ICGFLTU 寄存器不会影响 FLT。

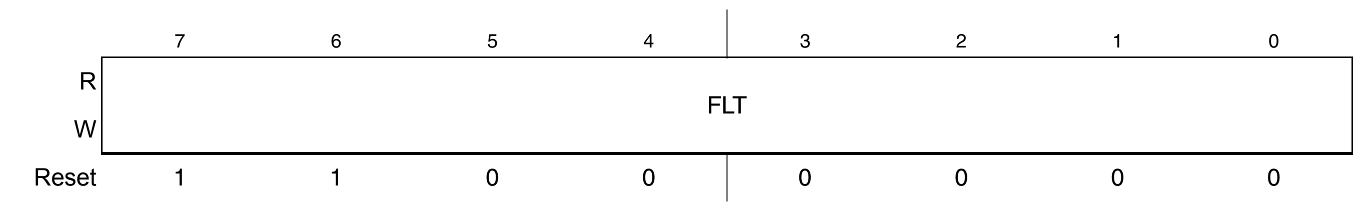


图 10-11. ICG 滤波寄存器低八位 （ICGFLTL）

表 10-6. ICGFLTL 寄存器位描述

位	描述
7:0 FLT	滤波器值—表明当前滤波器值，该值控制 DCO 频率。除了自时钟模式 （CLKS = 00），该位可读。在自时钟模式下任何写 ICGFLTU 将更新当前 12 位滤波器值。如果前一个关锁序列未完成，写 ICGFLTU 寄存器不会影响 FLT。 滤波器寄存器显示滤波器值 （FLT）

10.4.6 ICG 调整寄存器 （ICGTRM）

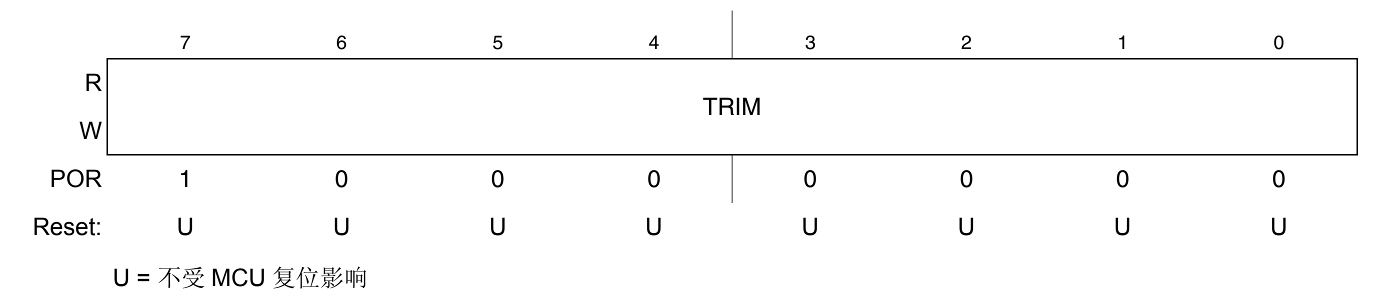


图 10-12. ICG 调整寄存器 （ICGTRM）

表 10-7. ICGTRM 寄存器位描述

位	描述
7：0 TRIM	ICG 调整设置—该位控制内部参考发生器频率。允许 POR 周期有 25% 浮动。该位通过二进制加权改变周期（例如，位 1 调整是位 0 调整的两倍）。在调整时，增加其二进制值将会增长周期，反之，减少值将缩短周期。

10.5 功能描述

本节将描述 ICG 五个工作模式的功能。同时讨论时钟丢失和失锁错误及进入每个模式的必要条件。 ICG 非常具有灵活性，在某些配置下它可能超越某个时钟规定。当使用 FLL 时，配置 ICG，则 ICGDCLK 的频率不会超出它的最大值，确保 MCU 正常工作。

10.5.1 OFF 模式

通常情况下，当 CPU 进入停止模式时， ICG 将停止所有时钟并进入 OFF 模式。当然，在 STOP 模式下，有两种情况时钟继续工作。

10.5.1.1 活跃 BDM

当 BDM 允许时， ICG 继续根据最初设置运行。这将允许通过 BDC 控制器访问存储器和控制寄存器。

10.5.1.2 OSCSTEN 位设置

当在停止模式下允许振荡器时 （OSCSTEN=1），除了单独的时钟发生器工作外， MCU 其他部分所需的时钟源都被关闭。这种方式避免了冗长的振荡器启动时间，或在停止模式 3 下使用振荡器来运行 RTI。

10.5.1.3 停止 /OFF 模式恢复

如果中断从停止模式唤醒 CPU，则先前设置的控制位有效，且系统时钟恢复工作。如果是 FEE 模式，ICG 将以内部参考时钟为源时钟直到外部时钟稳定。如果是 FBE 模式，在使能 ICGOUT 之前 ICG 将等待外部时钟稳定。

如果由复位从停止模式唤醒 CPU，先前设置的 ICG 控制位将被忽略，而使用缺省的复位值。因此退出 Stop 停止模式后， ICG 进入 SCM 模式，并被配置为一个大约 8MHz 的 DCO 输出。如果使用外部晶振，在使能 ICGERCLK 之前，将先检测 4096 个时钟信号。这包含在晶振的启动时间里。

10.5.2 自时钟模式 （SCM）

自时钟模式 （SCM）是缺省的操作模式，当以下条件中任何一个发生时就将进入 SCM 模式：

- 任何复位。
- 当 CLKS 不等于 10 时，从 OFF 模式退出。如果 CLKS = X1， ICG 将临时进入该状态直到 DCO 稳定（DCOS=1）。
- CLKS 由 X1 改写到 00
- CLKS = 1X 且 ICGERCLK 未被检测到 （ERCS = 0、 LOCS = 1）。

在这模式下， FLL 环路是敞开的。 DCO 正在运行，其输出时钟信号 ICGOUT 频率为 $f_{ICGDCLK}$ /R。ICGDCLK 频率在 8 MHz 到 40 MHz 之间变化，由滤波器寄存器 （ICGFLTH 和 ICGFLTL）决定。滤波器寄存器仅在此模式下可写。

如果由于复位进入此模式， $f_{ICGDCLK}$ 将缺省为 8 MHz。如果从 FEI 模式进入此模式， $f_{ICGDCLK}$ 将保持先前的频率，但如果从 FEE 模式进入 (不管是设置 CLKS，还是外部参考信号丢失)， $f_{ICGDCLK}$ 将保持先前的频率，而如果 FLL 失锁， ICGOUT 将变为其两倍。如果从 OFF 模式 (OFF) 进入此模式，在进入 OFF 模式前 $f_{ICGDCLK}$ 将等于 ICGDCLK 的频率。如果在退出 OFF 模式时 CLKS 位被置为 01 或 11，那么直到 ICGDCLK 稳定 (DCOS = 1)， ICG 才进入自时钟模式。在 ICGDCLK 稳定后， ICG 自动关闭环路，并切换到 FLL 参与的模式，由 CLKS 决定内部信号或外部信号使用。

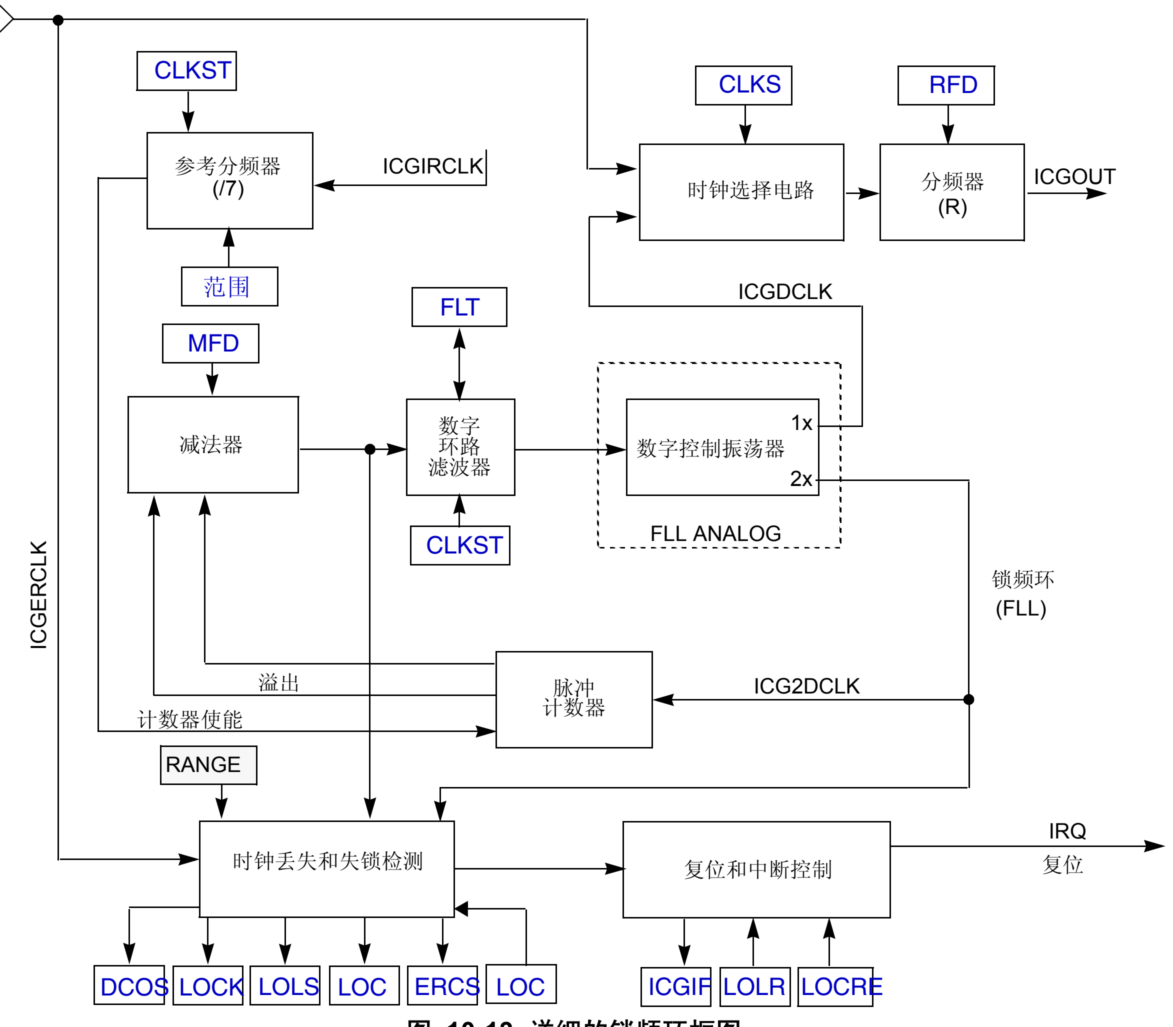


图 10-13. 详细的锁频环框图

10.5.3 FLL 内部时钟模式 （FEI）

当下列任一条件发生时进入 FEI 模式：

- CLKS 被写为 01
- 当退出 OFF 模式时， CLKS = 01。并进入自时钟模式， DCO 时钟稳定 （DCOS = 1）

在 FEI 模式下，采用内部时钟 ICGIRCLK 做为参考时钟源，该 FLL 环路试图锁定 ICGDCLK 频率在 MFD 设定的预期值。

10.5.4 FLL 内部未锁定

FEI 未锁定状态是一种临时状态，当计数偏差大于 n_{unlock} 最大值或小于 n_{unlock} 最小值， ICG 将进入并保持这种状态。状态由锁状态检测器检测。

此时， ICGOUT 频率等于 $f_{ICGDCLK}$ /R。

10.5.5 FLL 内部锁定

对于给定数目的样本，如果减法器的计数误差小于 n_{unlock} 最大值且大于 n_{unlock} 最小值，将从 FEI 失锁进入本状态。此时时钟信号输出 ICGOUT 频率等于 $f_{ICGDCLK}$ /R。在锁定的 FEI 模式下，滤波器值仅在每四个比较周期更新一次。更新的值为前四次比较取样误差的平均值。

10.5.6 FLL 旁路外部时钟模式 （FBE）

当下列任何一种条件发生时进入 FLL 旁路外部时钟模式：

- 当 CLKS = 10 且 ERCS 为高时从 SCM 进入
- 当 CLKS = 10 时， ERCS = 1 进入 OFF 模式，然后退出 OFF 模式
- 如果 DCO 时钟丢失，而外部时钟参考保持有效 （LOCS = 1、 ERCS = 1）从 FLL 外部时钟模式进入

在这种模式下， DCO 和 IRG 关闭，参考时钟为外部参考时钟 ICGERCLK。该输出时钟信号 ICGOUT 频率为 $f_{ICGDCLK}$ /R。如果使用外部时钟源 （REFS = 0），那么 EXTAL 引脚上的输入频率范围：0 MHz ～ 40 MHz。如果使用晶振或振荡器，那么频率范围可为低频范围 （RANGE = 0）或为高频范围 (RANGE = 1)。

10.5.7 FLL 外部时钟模式 （FEE）

当下列任何一种条件发生时进入 FEE 模式：

- CLKS = 11 并且 ERCS 和 DCOS 都为高
- 当退出 OFF 模式时， CLKS = 11。并进入自时钟模式， DCO 时钟稳定 （DCOS = 1）

在 FEE 模式下，参考时钟使用外部参考时钟 ICGERCLK，同时 FLL 环路试图锁定 ICGDCLK 频率以达到预设值。为了在 FEE 模式下运行，必须有一个 32 kHz ～ 100 kHz 或 2 MHz ～ 10 MHz 的外部时钟源。在 FEE 模式下外部时钟频率的最大值被限制在 10 MHz，以防止 DCO 溢出。 FLL 的最小倍频因子从表 10-12 可知是 4。因为 4 × 10 MHz 是 40 MHz，这是 DCO 的运行的极限，所以参考时钟不能超过 10 MHz。

10.5.7.1 FLL 外部失锁

当进入 FEE 模式，对于给定数目的样本，减法器的计数误差小于 n_{unlock} 最大值且大于 n_{unlock} 最小值，将进入失锁的 FEE 模式。

减法器的计数误差小于 n_{unlock} 最大值且大于 n_{unlock} 最小值， ICG 将一直处于本模式。

在这种状态下，脉冲计数器、减数器、数字环路滤波器，以及 DCO 组成一个回路，根据其运行条件 (本章后面的部分将进行讲解) 试图进行锁定。从进入该状态，到锁定 FLL， ICGOUT 的频率为 $f_{ICGDCLK}$ / （2 × R）。这额外的除 2，可以防止在锁定过程中，频率超过范围。在成功锁定 FLL 后，如果在 FEE 模式下，发生意外失锁，而导致重新进入该状态，此时 ICGOUT 的输出频率为 $f_{ICGDCLK}$ /R。

10.5.7.2 FLL 外部锁定

对于给定数目的样本，当减法器的计数误差小于 n_{unlock} 最大值且大于 n_{unlock} 最小值，就从 FEE 未锁定模式进入 FEE 锁定模式。此时时钟信号输出 ICGOUT 频率为 $f_{ICGDCLK}$ /R。在 FLL 外部锁定模式，仅在每四个比较周期，采用前四次比较取样误差的平均值更新滤波器。

10.5.8 FLL 锁定和失锁检测

为决定 FLL 锁定和失锁条件，脉冲计数器累加 DCO 一个比较周期的脉冲数 （见表 10-9 比较周期的解释），并把该值传给减法器。减法器把它与 MFD 中的值做比较，产生一个计数误差， Δn。为达到锁定状态，

Δn 必须在 n_{unlock} 最大值与最小值之间。在 FLL 锁定后，Δn 必须处在 n_{unlock} 最大值与最小值之间以保持锁定。如果 Δn 意外溢出此范围， LOLS 状态位被置位并保持置位直到通过软件清零或 MCU 复位。 LOLS 清除方法 : 读 ICGS1 然后写 1 给 ICGIF （LOLRE=0），或通过一个时钟丢失引起复位，或通过任意 MCU 复位。

当 ENBDM = OSCSTEN = 0 时，如 ICG 由停止模式进入 OFF 模式，则进入 FLL 失锁状态 （LOCK 被清零），但 LOLS 保持不变因为这不是一个意外的失锁条件。如果在停止模式下， ENBDM 被清为 0， ICG 进入 OFF 模式，尽管这不太常见。因为这是一个意外的时钟停止，当 MCU 从停止模式唤醒时 LOLS 将被置位。

当 MFD 或 CLKS 位被改变或在 FEI 模式下时，当 TRIM 位被改变时，将进入失锁状态。在这些情况下，LOCK 位将被清零直到 FLL 恢复锁定，但 LOLS 将不被置位。

10.5.9 FLL 时钟丢失检测

监控参考时钟和 DCO 时钟的条件不同 （见表 10-8）。假定参考频率正被监控， ERCS = 1 表明参考时钟达到了最小频率值要求。当参考时钟和 / 或 DCO 时钟正被监控时，如果它们中的任一个下降到某一频率下（分别为 f_{LOR} 和 f_{LOD} ）， LOLS 将通过被置位来指示错误。 LOCS 将保持置位直到它被清除或直到 MCU 复位。通过读 ICGS1 然后写 1 给 ICGIF （LOCRE=0）可清除 LOCS，或通过时钟丢失引起复位，或通过任意 MCU 复位。

如果 ICG 处于 FEE 模式，参考时钟的丢失将导致 ICG 进入 SCM 模式， DCO 时钟丢失将导致 ICG 进入 FBE 模式。如果 ICG 处于 FBE 模式下，参考时钟的丢失将导致 ICG 进入 SCM 模式。在上述情况下，CLKST 和 CLKS 位将被自动改变以反映当前状态。

如果时钟丢失发生在 FEE 模式，同时 ERCS 仍为 1，那么 CLKST 将被设置为 10，同时 ICG、转换到 FBE 模式。

在 FEE 或 FEI 模式，时钟丢失将导致失锁。因为清 LOCS 和 LOLS 位的方法相同，极少出现的 LOLRE = 1 但是 LOCRE = 0 将是一个问题。在这种情形下，锁丢失引起的复位中断将被忽略。

表 10-8. 时钟监控 （当 LOCD = 0 时）

模式	CLKS	REFST	ERCS	外部参考时钟监控？	DCO 时钟监控？
OFF 模式	0X 或 11	X	强制低	无	无
	10	0	强制低	无	无
	10	1	实时[1]	有 [1]	无
SCM 模式（CLKST=00）	0X	X	强制低	无	有[2]
	10	0	强制高	无	有 [2]
	10	1	实时	有	有 [2]
	11	X	实时	有	有 [2]
FEI（CLKST=01）	0X	X	强制低	无	有
	11	X	实时	有	有

表 10-8. 时钟监控 （当 LOCD = 0 时）(续)

模式	CLKS	REFST	ERCS	外部参考时钟监控?	DCO 时钟监控?
FBE（CLKST=10）	10	0	强制高	无	无
	10	1	实时	有	无
FEE (CLKST=11)	11	X	实时	有	有

[1] 如果 ENABLE 为高 （在退出停止模式后等待外部晶振启动）。
[2] 从 OFF 模式或 FLL 旁路外部模式进入 SCM 模式，在 DCO = 1 之前， DCO 时钟将不被监控。

10.5.10 时钟模式必要条件

通过写 CLKS1:CLKS0 请求时钟模式，同时 CLKST1:CLKST0 指示当前的时钟模式。如果满足最小条件， CLKST1:CLKST0 指示的状态应该和 CLKS1:CLKS0 中请求的一致。表 10-9 列出了 CLKS 和 CLKST 之间的关系。它也列出了 CLKS = CLKST 的条件或 CLKS ≠ CLKST 的原因。

注意

如果在下次复位之前需要使用晶振，在第一次写 ICGC1 寄存器前，必须设置 REFS = 1 和 CLKS = 1x。否则，将导致锁定 REFS = 0，这将在下一次复位前禁用振荡放大器。

表 10-9. ICG 状态表

实际模式 CLKST	期望的模式 CLKS	范围	参考频率（$f_{REFERENCE}$）	比较周期时间	ICGOUT	CLKS = CLKST 的条件[1]	CLKS1 ≠ CLKST 的原因
OFF 模式 (XX)	关 (XX)	X	0	—	0	—	—
	FBE(10)	X	0	—	0	—	ERCS = 0
SCM (00)	SCM(00)	X	$f_{ICGIRCLK}$ /7[2]	8/$f_{ICGIRCLK}$	ICGDCLK/R	不从 FBE 跳转到 SCM	—
	FEI(01)	0	$f_{ICGIRCLK}$ /7[1]	8/$f_{ICGIRCLK}$	ICGDCLK/R	—	DCOS = 0
	FBE(10)	X	$f_{ICGIRCLK}$ /7[1]	8/$f_{ICGIRCLK}$	ICGDCLK/R	—	ERCS = 0
	FEE(11)	X	$f_{ICGIRCLK}$ /7[1]	8/$f_{ICGIRCLK}$	ICGDCLK/R	—	DCOS = 0 或 ERCS = 0
FEI(01)	FEI(01)	0	$f_{ICGIRCLK}$ /7	8/$f_{ICGIRCLK}$	ICGDCLK/R	DCOS=1	—
	FEE(11)	X	$f_{ICGIRCLK}$ /7	8/$f_{ICGIRCLK}$	ICGDCLK/R	—	ERCS = 0

表 10-9. ICG 状态表 （续）

实际模式 CLKST	期望的模式 CLKS	范围	参考频率 （$f_{REFERENCE}$）	比较周期时间	ICGOUT	CLKS = CLKST 的条件[1]	CLKS1 ≠ CLKST 的原因
FBE (10)	FBE(10)	X	0	—	ICGDCLK/R	ERCS=1	—
	FEE(11)	X	0	—	ICGDCLK/R	—	LOCS = 1 & ERCS = 1
FEE (11)	FEE (11)	0	$f_{ICGIRCLK}$	2/ $f_{ICGIRCLK}$	ICGDCLK/R[3]	ERCS = 1 且 DCOS = 1	—
		1	$f_{ICGIRCLK}$	128/ $f_{ICGIRCLK}$	ICGDCLK/R[2]	ERCS = 1 且 DCOS = 1	—

[1] 写 CLKS 后 CLKST 不会立即更新。 CLKST 更新到新值需要几个总线周期。

[2] 在 SCM 模式下参考频率对 ICGOUT 无效，但该参考频率仍然用在作比较，其比较误差决定 DCOS 位。

[3] 在初始锁定过程中， $f_{ICGIRCLK}$ 为 ICGDCLK/2R。当 MFD 位被改变后， FLL 重新锁定，此时 $f_{ICGIRCLK}$ 为 ICGDCLK/R。

10.5.11 固定频率时钟

ICG 提供一个固定频率时钟输出 XCLK 用于片上外设。在除 FEE 以外的所有模式下，该时钟输出等于内部总线时钟 BUSCLK。在 FEE 模式下，如满足下列条件， XCLK 等于 ICGERCLK 除以 2：

- （P×N） /R ≥ 4， P 由 RANGE 决定 （见表 10-11）， N 和 R 分别由 MFD 和 RFD 决定 （见表 10-12）。
- LOCK = 1

如果以上两个条件不能满足，那么 XCLK 等于 BUSCLK。

当 ICG 在 FEI 或 SCM 模式时， XCLK 被关闭，任何使用 XCLK 作为时钟源的外设不能使用 XCLK。

10.5.12 高增益振荡器

振荡器可以运行在高增益振荡器 （HGO）模式，当在 FBE 或 FEE 模式下时，高增益振荡器 （HGO）模式能提高振荡器对 EMC 噪声的抵抗力。通过写 1 到 ICGC1 寄存器中的 HGO 位使能该模式。 HGO 可与高频和低频振荡器一起使用，但仅当 ICGC1 寄存器中的 REFS = 1 时有效。当 HGO = 0 时，选择的是标准低功耗振荡器。复位后该位只能写一次。

10.6 初始化 / 应用信息

10.6.1 概述

该节给出了初始化时的基本指导，以及用户可参考的配置。在一些应用中，串行通信连接需要准确的参考时钟。在另一些应用中，最低功耗可能是时钟设置主要考虑的事项。也有一些应用，低成本是最主要的目标。 ICG 具有很大的灵活性，为不同的应用选择最佳的时钟。

表 10-10. ICG 配置原则

	时钟参考源 = 内部	时钟参考源 = 外部
使用 FLL	FEI 4 MHz < f_{Bus} < 20 MHz 中等功率 （如果震荡范围 = 高，将会小于 FEE） 较高的时钟精确度 （在 IRG 调整后） 最低系统成本 （不需要外部器件） IRG 打开。 DCO 打开[1]	FEE 4 MHz < f_{Bus} < 20 MHz 中等功率 （如果震荡范围 = 低，将会小于 FEI） 高时钟精确度 中等或高系统成本 （需要晶振，共鸣器或外部时钟源器件） IRG 关闭。 DCO 打开
FLL 旁路	**SCM** 该模块主要提供给迅速和可靠的系统启动。 3 MHz < f_{Bus} < 5 MHz （缺省） 3 MHz < f_{Bus} < 20 MHz （通过过滤器位） 中等功率 低精确度 IRG 关闭。 DCO 打开并且开环。	**FBE** 当使用晶振或蜂鸣器时， $f_{Bus} \leq 8$ MHz。 最低功率 最高时钟精确度 中等或高系统成本 （需要晶振，共鸣器或外部时钟源器件） IRG 关闭。 DCO 关闭。

[1] IRG 通常消耗 100 µA。根据输出频率， FLL 和 DCO 通常消耗 0.5 ～ 2.5 mA。为了最小功耗和最小抖动，选择尽可能小的 N 和 R。

下面将根据不同配置举例介绍。

注意

前缀 $ 表示十六进制，前缀 % 表示二进制，无前缀表示十进制。

重要的配置信息在这里列出，以供参考。

表 10-11. ICGOUT 频率计算选项

时钟配置	f_{ICGOUT}[1]	P	注意
SCM	$f_{ICGDCLK}$ /R	NA	复位后通常 f_{ICGOUT} = 8 MHz
FBE	f_{ext} /R	NA	
FEI	f_{ICG} /R	64	通常 f_{IRG} = 243 MHz
FEE	f_{ICGOUT} *P*N/R	范围 = 0； P = 64 范围 = 1； P = 1	

[1] 确保 $f_{ICGDCLK}$ 等于 f_{ICGOUT}*R，不要超过 $f_{ICGDCLKmax}$

表 10-12. MFD 和 RFD 编码表

MCF 值	倍增因子		RFD	分频因子
000	4		000	÷ 1
001	6		001	÷ 2
010	8		010	÷ 4
011	10		011	÷ 8
100	12		100	÷ 16
101	14		101	÷ 32
110	16		110	÷ 64
111	18		111	÷ 128

10.6.2 例 1：外部晶振 = 32 kHz，总线频率 = 4.19 MHz

在该例中， FLL 将倍频外部 32 kHz 振荡器到 8.38 MHz，来产生 4.19 MHz 的总线频率。

MCU 复位后， ICG 处于自时钟模式 （SCM），提供大约 8 MHz 的 ICGOUT，对应于 4 MHz 的总线频率（f_{Bus}）。

时钟配置是使用 FLL，外部 （FEE）。所以有：

$$f_{ICGOUT} = f_{ext} * P * N/R\ ;\quad P = 64,\quad f_{ext} = 32\ kHz \qquad \text{公式 10-1}$$

N/R：

$$N/R = 8.38\ MHz / (32\ kHz * 64) = 4\ ;\ \text{可以选择}\ N = 4,\ R = 1 \qquad \text{公式 10-2}$$

为执行操作，每个寄存器需要的值如下：

ICGC1 = $38 (%00111000)

Bit 7	HGO	0	配置振荡器低功耗
Bit 6	RANGE	0	配置振荡器低频率， FLL 预分频因子为 64
Bit 5	REFS	1	振荡器需要使用晶振或谐振器
Bits 4:3	CLKS	11	使用 FLL 外部参考时钟模式
Bit 2	OSCSTEN	0	禁止振荡器
Bit 1	LOCD	0	允许丢失时钟检测
Bit 0		0	未使用或保留；读为 0

ICGC2 = $00 (%00000000)

Bit 7	LOLRE	0	失锁时产生一个中断请求
Bits 6:4	MFD	000	设置 MFD 倍乘因子为 4
Bit 3	LOCRE	0	丢失时钟失产生一个中断请求
Bits 2:0	RFD	000	设置分频因子为÷ 1

ICGS1 = $xx

该位只读，除了清中断标志

ICGS2 = $xx

该位只读；执行任何时间临界任务前应该读 DCOS = 1

ICGFLTLU/L = $xx

仅在自时钟模式时需要； FLT 通过回路调整产生 8.38 MHz DCO 时钟

Bits 15:12 未使用 0000

Bits 11:0 FLT 不需要用户初始化

ICGTRM = $xx

Bits 7:0 TRIM 当调整内部振荡器时，只需要写。采用外部晶振时不使用。

图 10-14 为需要 ICG 初始化的三个条件的流程图。

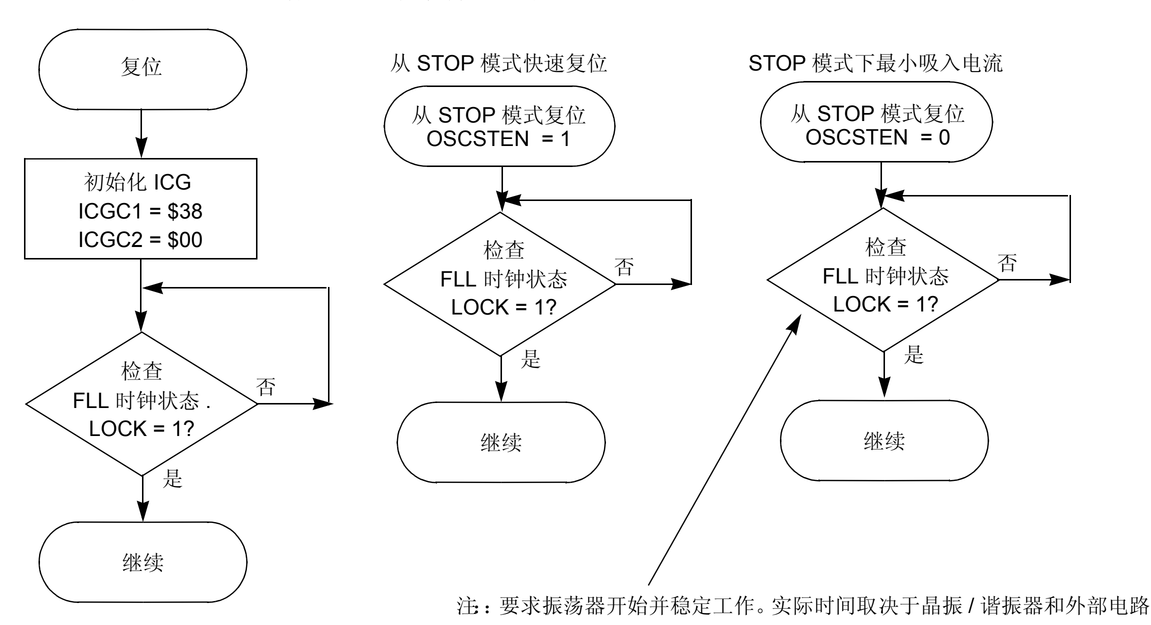


图 10-14. ICG 初始化用于 FEE 在例 1 情况下

10.6.3 例 2：外部晶振 = 4 MHz，总线频率 = 20 MHz

在该例中， FLL(在 FEE 模式) 将倍频外部 4 MHz 振荡器到 40 MHz，来产生 20 MHz 的总线频率。

MCU 复位后， ICG 处于自时钟模式 （SCM），提供大约 8 MHz 的 ICGOUT，对应于 4 MHz 的总线频率（f_{Bus}）。

复位初始化软件时，时钟配置为使用 FLL，外部 （FEE）。所以

$$f_{ICGOUT} = f_{ext} * P * N/R \;;\quad P = 1, f_{ext} = 4.00\ \text{MHz} \qquad \text{公式 10-3}$$

N/R：

$$N/R = 40\ \text{MHz} / (4\text{MHz} * 1) = 10\;;\ \text{可以选择}\ N = 10,\ R = 1 \qquad \text{公式 10-4}$$

为执行操作，每个寄存器需要的值如下：

ICGC1 = $78 (%01111000)

Bit 7	HGO	0	配置振荡器低功耗
Bit 6	RANGE	1	配置振荡器高频率； FLL 预分频因子为 1
Bit 5	REFS	1	振荡器需要使用晶振或谐振器
Bits 4:3	CLKS	11	使用 FLL 外部参考时钟模式
Bit 2	OSCSTEN	0	禁止振荡器
Bit 1	LOCD	0	允许丢失时钟检测
Bit 0		0	保留或未使用；读为 0

ICGC2 = $30 (%00110000)

Bit 7	LOLRE	0	失锁时产生一个中断请求
Bits 6:4	MFD	011	设置 MFD 倍乘因子为 10
Bit 3	LOCRE	0	丢失时钟产生一个中断请求
Bits 2:0	RFD	000	设置分频因子为 ÷ 1

ICGS1 = $xx

该位只读，除了清中断标志

ICGS2 = $xx

该位只读；执行任何时间临界任务前应该读 DCOS = 1

ICGFLTLU/L = $xx

在该例中未使用

ICGTRM = $xx

在该例中未使用

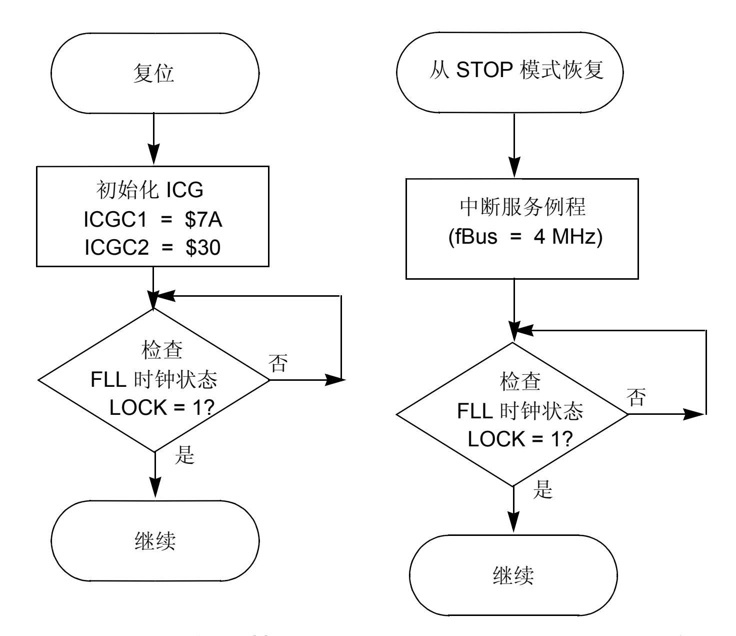


图 10-15. 用于例 2 情况下的 ICG 初始化和从 stop 模式复位

10.6.4 例 3：无外部晶振，总线频率 = 5.4 MHz

在本例中， FLL 将倍频内部 243 kHz （大约）参考时钟到 10.8 MHz，来产生 5.4 MHz 的总线频率。该系统使用微调功能基于外部参考信号获得更精确的频率。

MCU 复位后， ICG 处于自时钟模式 （SCM），提供大约 8 MHz 的 ICGOUT，对应于 4 MHz 的总线频率（f_{Bus}）。

时钟配置是使用 FLL，内部 （FEI）。所以

$$f_{ICGOUT} = (f_{IRG} / 7) * P * N/R \text{；} P = 64, f_{IRG} = 243 \text{ kHz}$$ *公式 10-5*

N/R：

$$N/R = 10.8 \text{ MHz} / (243/7 \text{ kHz} * 64) = 4.86 \text{；可以选择 } N = 10, R = 2$$ *公式 10-6*

需要一个调整程序调整频率到正好 5.4 MHz。例 4 是一个调整程序的例子。

为执行操作，每个寄存器需要的值：

ICGC1 = $28 (%00101000)

Bit 7	HGO	0	配置振荡器低功耗
Bit 6	RANGE	0	配置振荡器低频率； FLL 预分频因子为 64
Bit 5	REFS	1	振荡器需要使用晶振或谐振器 （不使用该位）
Bits 4:3	CLKS	01	使用 FLL 内部参考时钟模式
Bit 2	OSCSTEN	0	禁止振荡器
Bit 1	LOCD	0	允许丢失时钟检测
Bit 0		0	保留或未使用；读为 0

ICGC2 = $31 (%00110001)

Bit 7	LOLRE	0	失锁时产生一个中断请求
Bits 6:4	MFD	011	设置 MFD 倍乘因子为 10
Bit 3	LOCRE	0	丢失时钟产生一个中断请求
Bits 2:0	RFD	001	设置分频因子为÷ 2

ICGS1 = $xx

该位只读，除了清中断标志

ICGS2 = $xx

该位只读；执行任何时间临界任务前建议读该位。

ICGFLTLU/L = $xx

在本例中未使用

ICGTRM = $xx

Bits 7:0 TRIM 当调整内部振荡器时，只需要写。独立操作 （参见例 4）。

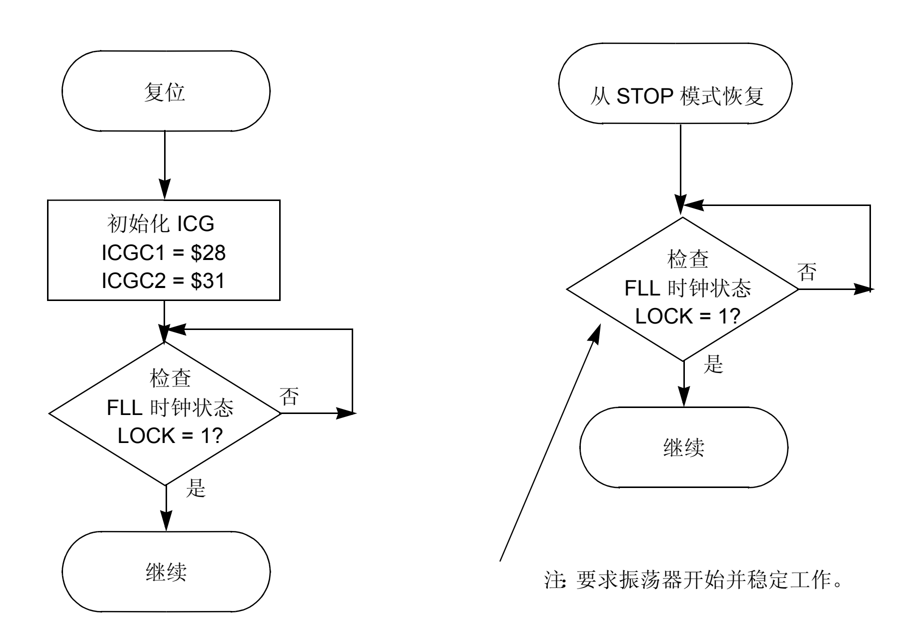


图 10-16. 用于例 3 情况下的 ICG 初始化和从 stop 模式复位

10.6.5 例 4：内部时钟产生器调整

内部产生的时钟源有一个 ±25% 的浮动值。在一些情况，此精度能够满足要求。但在一些要求频率精度很高的应用中，需要调整程序帮助产生更准确的时钟源。该节给出了一个调整内部振荡器的例子。除此外，还有很多其他调整的方法可供使用。

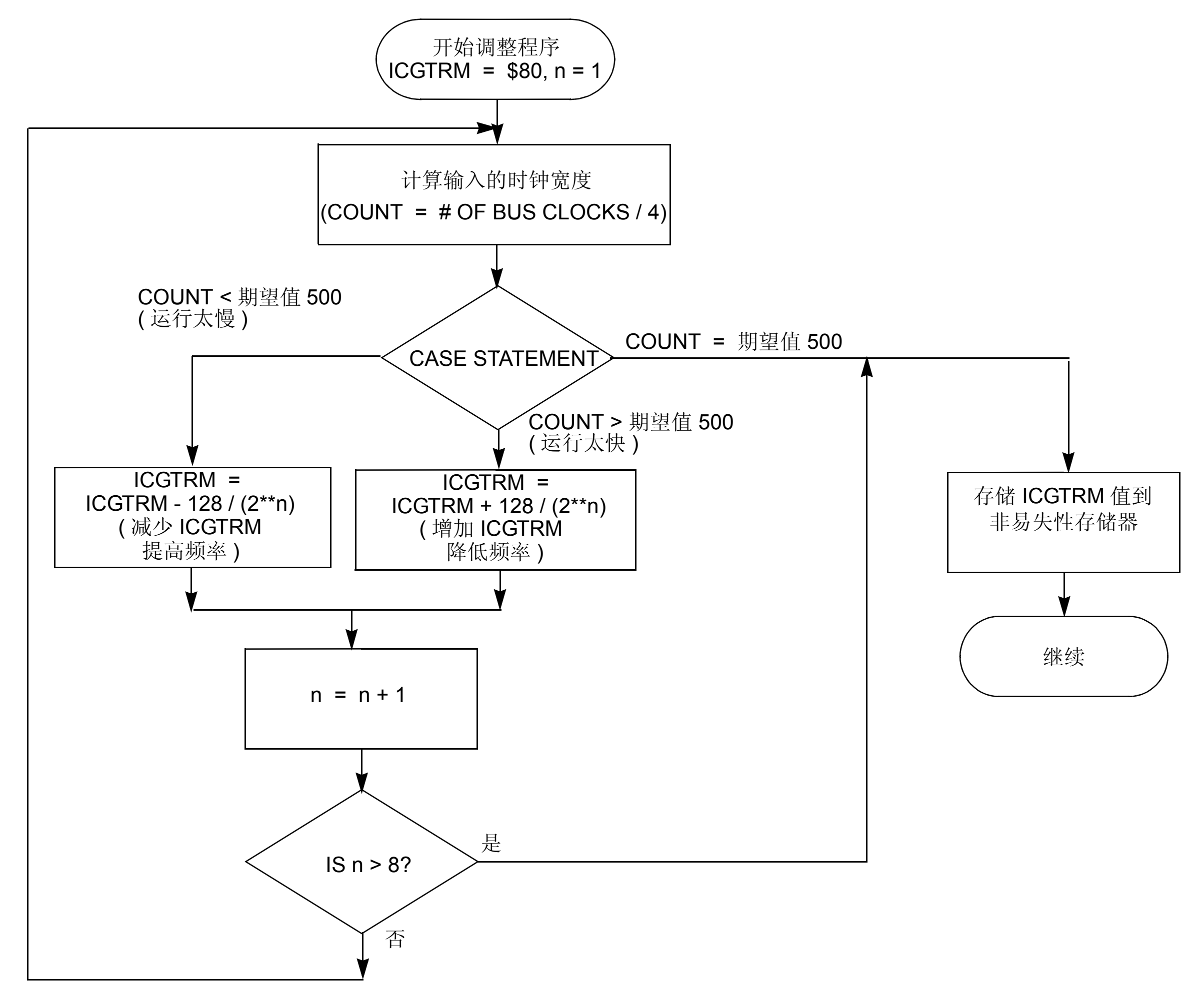


图 10-17. 调整程序

在这个特殊情况中， MCU 贴片到 PCB 上，整个装配正在用自动化测试工具进行最终测试。当用户使用软件控制时，一个单独的信号或信息提供给 MCU 操作。当测试者提供了一个准确的参考信号， MCU 根据图 10-17 流程所示，初始化调整程序。

如果希望总线频率接近设备所允许的最大值，推荐使用两倍于最终值的分频因子 （R）。调整程序完成后，恢复分频因子。这可以避免意外地超过最大时钟频率。

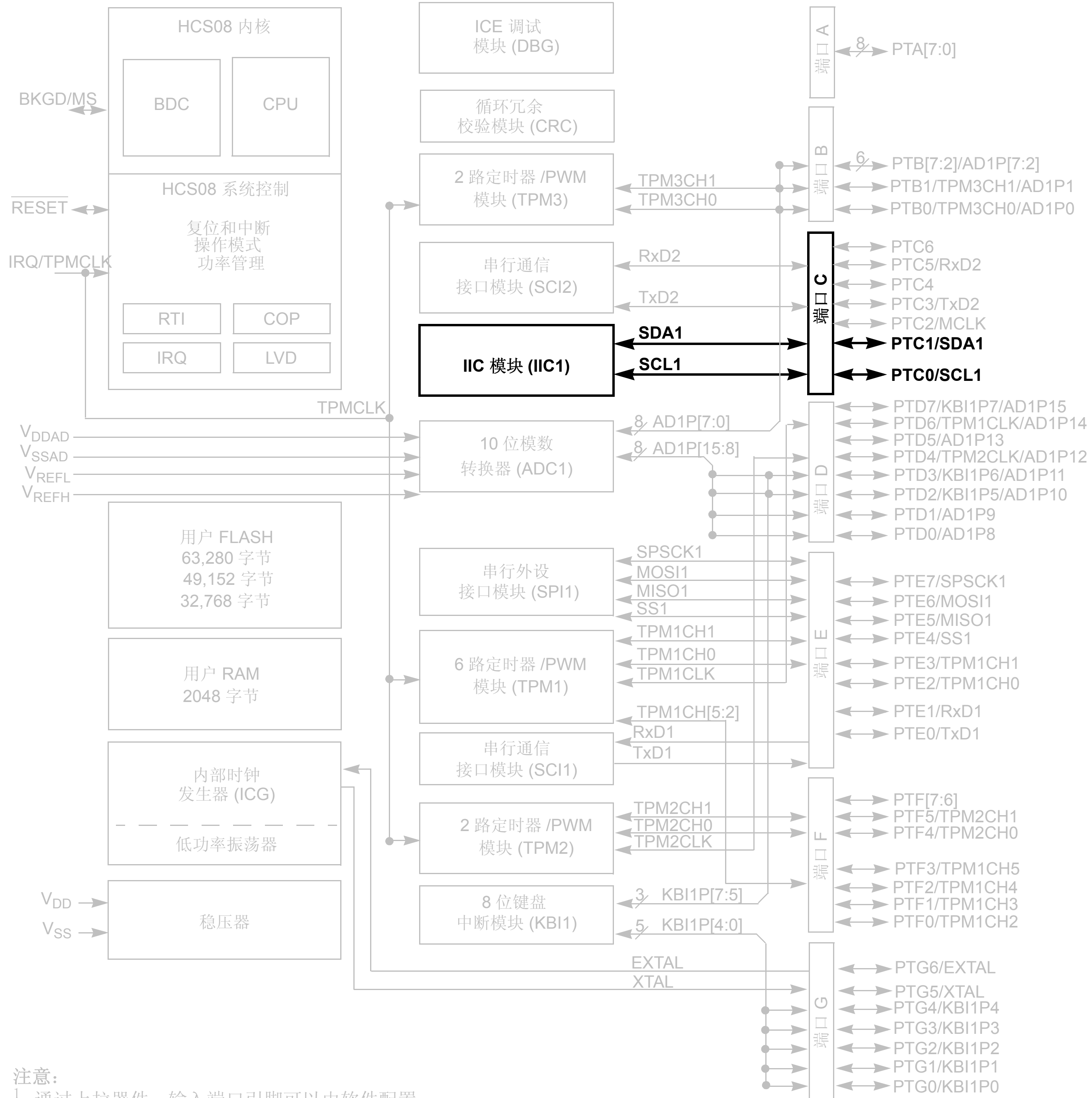
第 11 章
IIC 模块 (S08IICV2)

11.1 介绍

IIC 提供了不同器件间的通信方法。这个接口可以在最大的总线负载和时序下，支持最高 100 kbps 的传输速率。器件可以在较低总线负载下、以更高的波特率 （最高时钟 /20）运行。最大通信长度和可以连接的器件数量受 400 pF 的最高总线电容限制。

如需其他详情，请参见 《HCS08 参考手册第 1 卷》(飞思卡尔半导体文档订购编号 HCS08RMv1/D).

MC9S08AC60 系列的微控制器具有一个 IIC 模块，用于与其他集成电路进行通信。



注意：

1 通过上拉器件，输入端口引脚可以由软件配置。

2 如果 IRQ 使能 (IRQPE = 1)，引脚包含软件可配置的上拉 / 下拉器件。如果选择上升边沿检测，则下拉使能 (IRQEDG = 1)。

3 引脚包含集成上拉器件。

4 PTD3、PTD2、PTD7 和 PTG4 既包含上拉也包含下拉器件。当 KBI 使能时 (KBIPEn = 1)，且选择了上升边沿 (KBEDGn = 1)，则下拉使能 。

5 TPMCLK、 TPM1CLK 和 TPM2CLK 选项可通过软件配置； 复位后， TPM1CLK、 TPM2CLK 和 TPMCLK 分别提供给 TPM1、 TPM2 和 TPM3。

图 11-1. 显亮 IIC 模块的 MC9S08AC60 系列模块图

11.1.1 特性

IIC 具有以下特性：

- 兼容 IIC 总线标准
- 支持多主控
- 软件可编程 64 种不同串行时钟频率
- 应答位软件可选择
- 中断驱动的按字节数据传送
- 仲裁从主机到从机自动模式转换丢失的中断
- 调用地址识别中断
- 开始和停止信号发生 / 检测
- 重复产生开始信号
- 识别位发生 / 检测
- 总线忙检测
- 一般调用识别
- 10 位地址扩展

11.1.2 操作模式

下面简述不同 MCU 模式下的 IIC：

- 运行模式 — 这是基本的操作模式。为了降低功耗，在此模式下，禁用本模块。
- 等待模式 — 当 MCU 处在等待模式，该模块继续运作，并能提供唤醒中断。
- 停止模式 —为了降低功耗，IIC 在停止模式 3 下是禁止的。停止指令不影响 IIC 寄存器状态。停止模式 2 和停止模式 1 复位该寄存器内容。

11.1.3 模块结构图

图 11-2 是 IIC 的模块结构图。

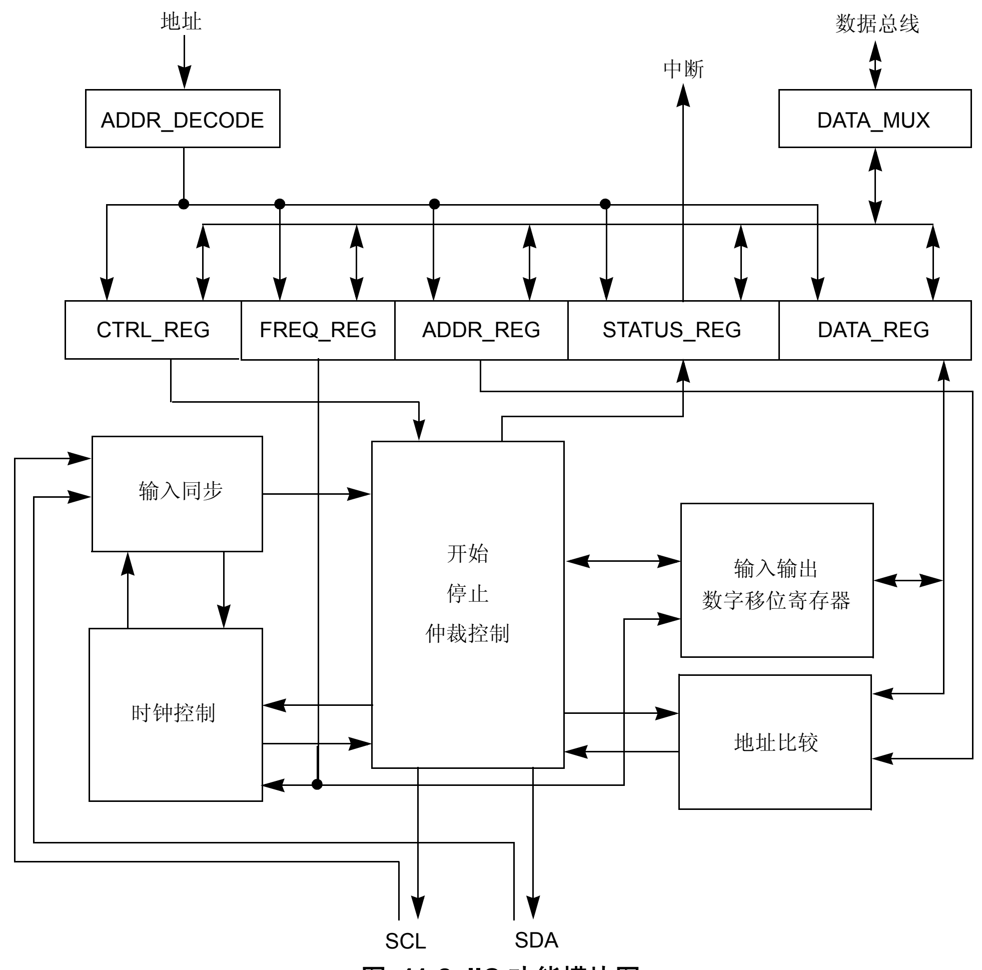


图 11-2. IIC 功能模块图

11.2 外部信号描述

本节介绍用户可访问的引脚信号。

11.2.1 SCL — 串行时钟线

该双向 SCL 是 IIC 系统的串行时钟线。

11.2.2 SDA — 串行数据线

该双向 SDA 是 IIC 系统的串行数据线。

11.3 寄存器定义

本节按地址顺序描述 IIC 寄存器。关于所有 IIC 寄存器的绝对地址分配，请参阅存储器章节的直接页寄存器摘要。本节只涉及到这些寄存器和控制位的名字。飞思卡尔提供头文件来把这些名字转换成相应的绝对地址。

11.3.1 IIC 地址寄存器 （IICA）

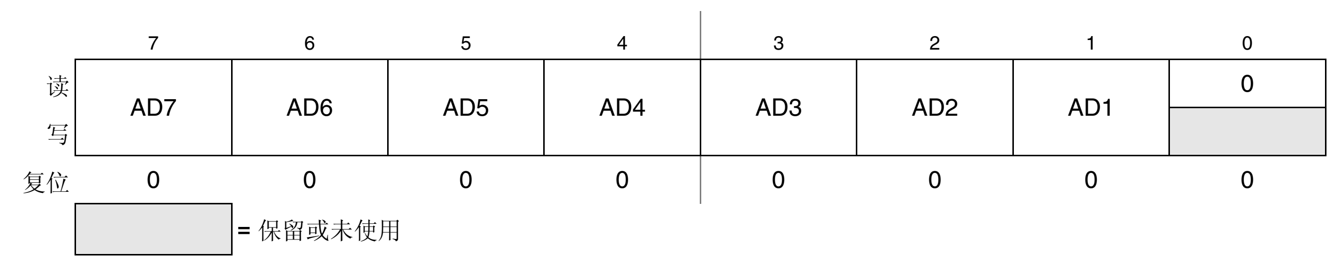


图 11-3. IIC 地址寄存器 （IICA）

表 11-1. IICA 位描述

位	描述
7-1 AD[7:1]	从机地址。 AD[7:1] 包含被 IIC 模块使用的从机地址。该位使用 7 位地址方案和 10 位地址方案中的低七位。

11.3.2 IIC 频率分频寄存器 （IICF）

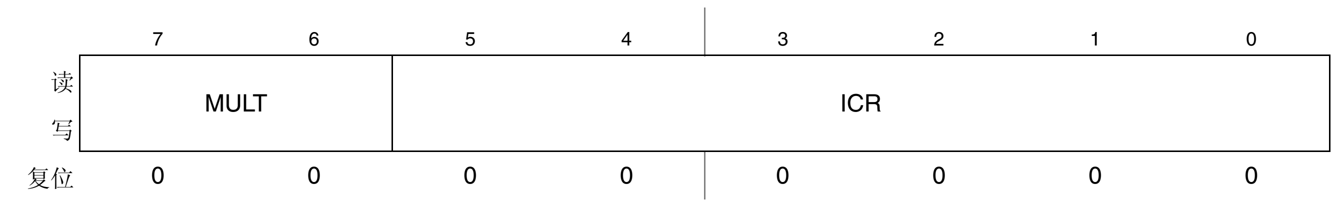


图 11-4. IIC 频率分频寄存器

表 11-2. IICF 位描述

位	描述
7-6 MULT	IIC 增频因子。 MULT 位定义了增频因子 mul，该因子和 SCL 分频因子一起，用于设置 IIC 的波特率。 Mul 定义如下： 00 mul = 01 01 mul = 02 10 mul = 04 11 保留
5-0 ICR	IIC 时钟速率。 ICR 被用来分频总线时钟用于位率选择。这些位和 MULT 一起用于确定 IIC 的波特率、 SDA 保持时间、 SCL 开始保持时间和 SCL 停止保持时间。 表 11-4 提供了 SCL 因子和保持值。 SCL 分频因子乘以 MULT 增频因子得到 IIC 波特率 IC 波特率 = 总线速度 /(倍频因子 * 分频因子) SDA 保持时间是从 SCL 的下降沿 （IIC 时钟）到 SDA 改变 （IIC 数据）这段时间。 SDA 保持时间 = 总线周期 *mul*SDA 保持值 SCL 开始保持时间是当 SCL 处于高 （开始条件），从 SDA 下降沿 （IIC 数据）开始，到 SCL 下降沿 (IIC 时钟) SCL 开始保持时间 = 总线周期 *mul*SCL 开始保持值 SCL 停止保持时间是从 SCL 上升沿 （IIC 时钟）开始到 SDA 上升沿，此时 SCL 为高 （停止条件）。 SCL 停止保持时间 =CPU 总线周期 *mul*SCL 停止保持值

例如，表 11-3 为总线速度为 8 MHz 时，不同的 ICR 和 MULT 组合获得 100 kbps 的 IIC 波特率的保持时间值。

表 11-3. 8 MHz 总线速度保持时间值

MULT	ICR	保持时间 （微秒）		
		SDA	SCL 开始	SCL 停止
0x2	0x00	3.500	3.000	5.500
0x1	0x07	2.500	4.000	5.250
0x1	0x0B	2.250	4.000	5.250
0x0	0x14	2.126	4.25.	5.125
0x0	0x18	1.125	4.750	5.125

表 11-4. IIC 分频因子和保持值

ICR （hex）	SCL 分频因子	SDA 保持值	SCL 保持值 （开始）	SCL 保持值 （停止）		ICR （hex）	SCL 分频因子	SDA 保持值	SCL 保持值 （开始）	SCL 保持值 （停止）
00	20	7	6	11		20	160	17	78	81
01	22	7	7	12		21	192	17	94	97
02	24	8	8	13		22	224	33	110	113
03	26	8	9	14		23	256	33	126	129
04	28	9	10	15		24	288	49	142	145
05	30	9	11	16		25	320	49	158	161
06	34	10	13	18		26	384	65	190	193
07	40	10	16	21		27	480	65	238	241
08	28	7	10	15		28	320	33	158	161
09	32	7	12	17		19	384	33	190	193
0A	36	9	14	19		2A	448	65	222	225
0B	40	9	16	21		2B	512	65	254	257
0C	44	11	18	23		2C	576	97	286	289
0D	48	11	20	25		2D	640	97	318	321
0E	56	13	24	29		2E	768	129	382	385
0F	68	13	30	35		2F	960	129	478	481
10	48	9	18	25		30	640	65	318	321
11	56	9	22	29		31	768	65	382	385
12	64	13	26	33		32	896	129	510	513
13	72	13	30	37		33	1024	129	510	513
14	80	17	34	41		34	1152	193	638	641
15	88	17	38	45		35	1280	193	574	577

表 11-4. IIC 分频因子和保持值 (续)

ICR（hex）	SCL 分频因子	SDA 保持值	SCL 保持值（开始）	SCL 保持值（停止）		ICR（hex）	SCL 分频因子	SDA 保持值	SCL 保持值（开始）	SCL 保持值（停止）
16	104	21	46	53		36	1536	257	766	769
17	128	21	58	65		37	1920	257	958	961
18	80	9	38	41		38	1280	129	638	641
19	96	9	46	49		39	1536	129	766	769
1A	112	17	54	57		3A	1792	257	894	897
1B	128	17	62	65		3B	2048	257	1022	1025
1C	144	25	70	73		3C	2304	385	1150	1153
1D	160	25	78	81		3D	2560	385	1278	1281
1E	192	33	94	97		3E	3072	513	1534	1537
1F	240	33	118	121		3F	3840	513	1918	1921

11.3.3 IIC 控制寄存器 （IICC1）

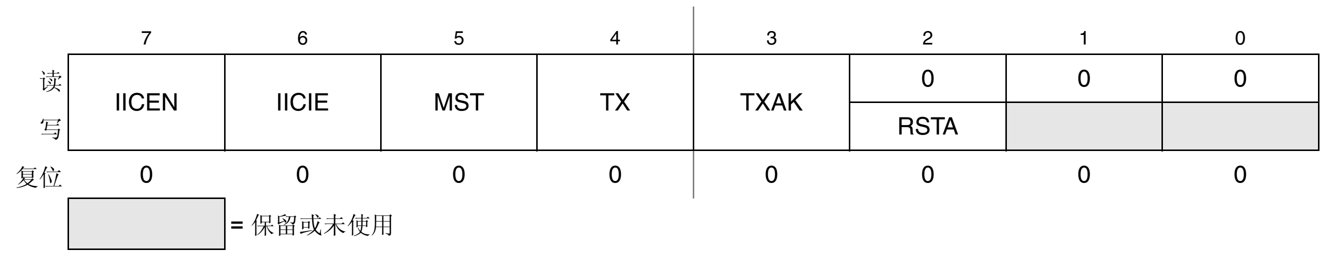


图 11-5. IIC 控制寄存器 （IICC1）

表 11-5. IICC1 位描述

位	描述
7 IICEN	IIC 使能位。该 IICEN 位决定 IIC 模块是否允许。 0 IIC 不允许 1 IIC 允许
6 IICIE	IIC 中断允许位。 IICIE 位决定 IIC 中断请求是否被允许。 0 IIC 中断请求不允许 1 IIC 中断请求允许
5 MST	主模式选作位。当总线上产生一个开始信号，并且 MCU 被确定为主模式时，该位从 0 变到 1。 当该位从 1 到 0， MCU 将在 IIC 总线上产生停止信号，操作模式从主模式变为从模式。 0 从机模式 1 主机模式

表 11-5. IICC1 位描述 (续)

位	描述
4 TX	发送模式选择位。该位用来选择主机和从机的数据传送方向。在主机模式下，该位必须根据数据传送类型来确定。所以，对于地址周期，本位总是为高。当 MCU 为从机时，该位必须根据状态寄存器的 SRW 位通过软件来设置。 0 接收模式 1 发送模式
3 TXAK	发送应答使能位。在主机和从机接受缓存的数据应答期间该位指定驱动到 SDA 上的值 0 在接受完一个字节后发送应答信号到总线 1 不发送应答信号
2 RSTA	重新开始。若 MCU 在主机模式下，写一个 1 到该位将产生一个重新开始信号。该位读出时总为 0。在错误的时间尝试重新开始将丢失仲裁。

11.3.4 IIC 状态寄存器 （IICS）

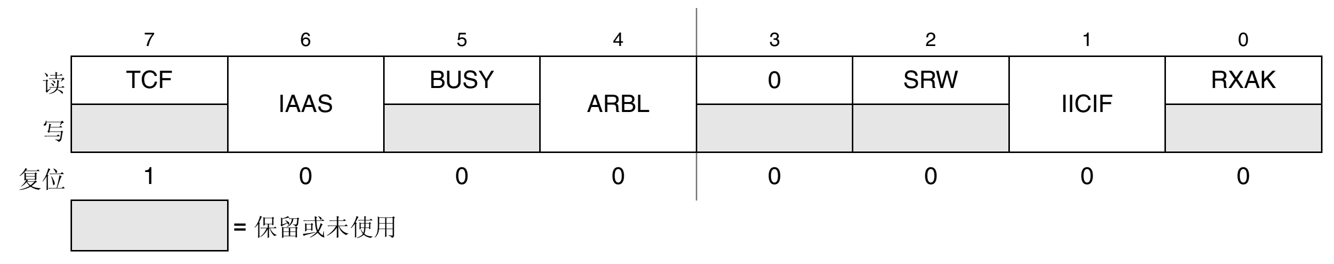


图 11-6. IIC 状态寄存器 （IICS）

表 11-6. IIC 位描述

位	描述
7 TCF	发送完成标识位。一个字节传送完毕后，该位将被置位。该位只在进行 IIC 模块传送时有效。在接收模式下，读 IIC 数据寄存器 IICD 或在发送模式下写 IICD 清除该位。 0 发送中 1 发送已完成
6 IAAS	地址被选择标志位。当 MCU 作为从机，并且它的地址与 IIC 总线上由主机发送的寻址地址吻合时，该位被置位。写 IICC 寄存器清零该位。 0 MCU 未被寻址 1 MCU 作为从机被主机寻址
5 BUSY	总线忙标志。该位表明总线的状态无论是在主机模式还是从机模式。当检测到开始信号，该位被置位。当检测到停止信号时，该位被清零 0 总线空闲 1 总线忙
4 ARBL	仲裁丢失标志位。当仲裁处理丢失，该位被硬件置位。必须通过软件的方法向该位写 1 来清除。 0 正常总线操作 1 仲裁丢失
2 SRW	从机读写标志位。当 MCU 作为从机被寻址时，该位为发送给主机的调用地址的读 / 写命令位的值。 0 从机为接收模式，主机向从机写数据 1 从机为发送模式，主机从从机读数据

表 11-6. IIC 位描述 (续)

位	描述
1 IICIF	IIC 中断标志位。当一个中断挂起时，该位被置位。必须通过软件方法来向该位写 1 清除该位。下列任一事件可以置位 IICIF：一个字节传送完毕；从地址符合调用地址，仲裁丢失。 0 无中断挂起 1 有中断挂起
0 RXAK	接收应答标识。当该位为 0 表明在总线上传送完一个字节后接收到了应答信号。如果为 1 则表示没有接收到应答信号 0 接收到应答信号 1 未接收到应答信号

11.3.5 IIC 数据输入输出寄存器 （IICD）

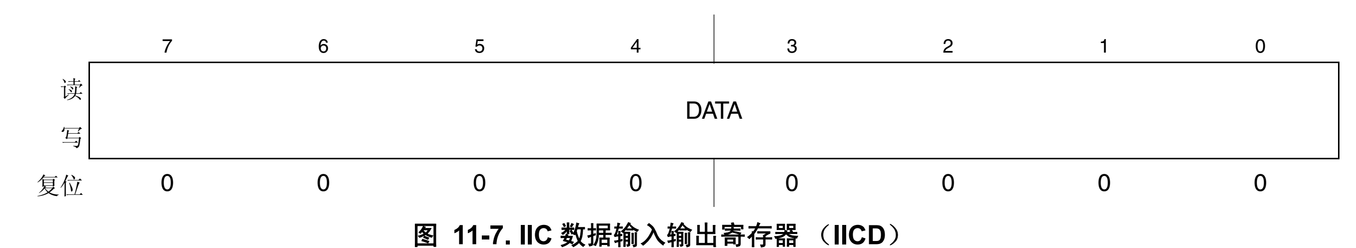


图 11-7. IIC 数据输入输出寄存器 （IICD）

表 11-7. IICD 位描述

位	描述
7-0 DATA	数据 — 在主机发送模式下，当数据写入 IICD 时，数据发送开始。先发送数据字节的最高位。在主机接收模式下，接收下个字节数据从读该寄存器开始。

注意

当 MCU 要从主接收模式变为主发送模式时，必须在读 IICD 寄存器之前进行 IIC 模式转换，以避免无意中初始化主机接收数据传送。

在从机模式下，地址匹配后，相同的功能也可用。

在主机和从机模式下传送一开始时， IICC 中的 TX 位必须正确反映传送目标方向。例如，如果 IIC 被配置为主机发送但一个主机接收被请求时，读 IICD 并不能开始接收。

当 IIC 被配置为主机或从机接收模式时，读 IICD 返回接收到最后一个字节数据。 IICD 并不影响在 IIC 总线上已经传送的每个字节，也不能通过读出已写入到 IICD 中的一个字节数据来校验此字节被正确写入。

在主机发送模式下，紧跟着 MST 断言被写到 IICD 的第一个字节，被用于发送地址，它应该包括与必需的 R/W 位 （位 0）相关的寻址地址 （位 7- 位 1）

11.3.6 IIC 控制寄存器 2 （IICC2）

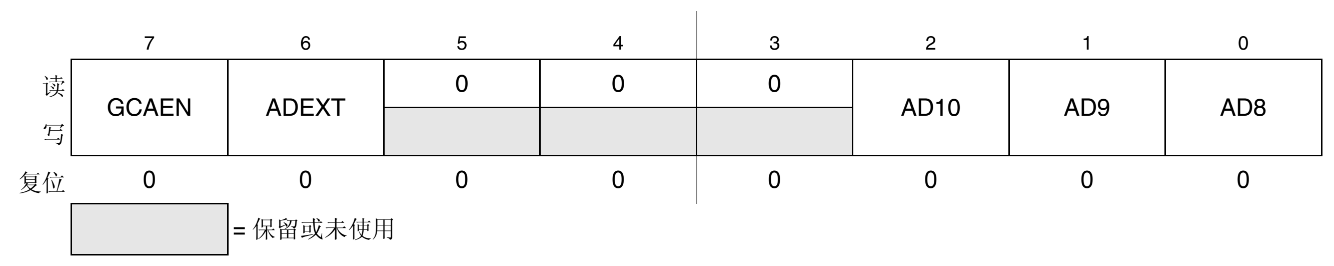


图 11-8. IIC 控制寄存器 2 （IICC2）

表 11-8. IICC2 位描述

位	描述
7 GCAEN	一般寻址使能位。 GCAEN 位允许或禁止一般寻址。 0 一般寻址禁止 1 一般寻址允许
6 ADEXT	地址扩展。 ADEXT 位控制用于从机地址的位数 0 7 位地址方案 1 10 位地址方案
2-0 AD[10:8]	从机地址。 AD[10:8] 为 10 位地址方案从机地址的高三位。仅当 ADEXT 被置位该位才有效。

11.4 功能描述

本节详细介绍 IIC 模块的功能。

11.4.1 IIC 协议

IIC 总线系统使用串行数据线 （SDA）和串行时钟线 (SCL) 来进行数据传送。所有连接到它的设备必须具有公开通道或公开搜集输出 （open collector output)。带有外部上拉电阻的两条线都可执行逻辑与操作。电阻值由系统决定。

通常情况下，标准的通信包含以下四个部分：

- 开始信号
- 从机地址传输
- 数据传输
- 停止信号

不要将停止信号和 CPU 的停止指令混为一谈。 IIC 总线系统通信将在下面进行简要描述，并附上图 11-9。

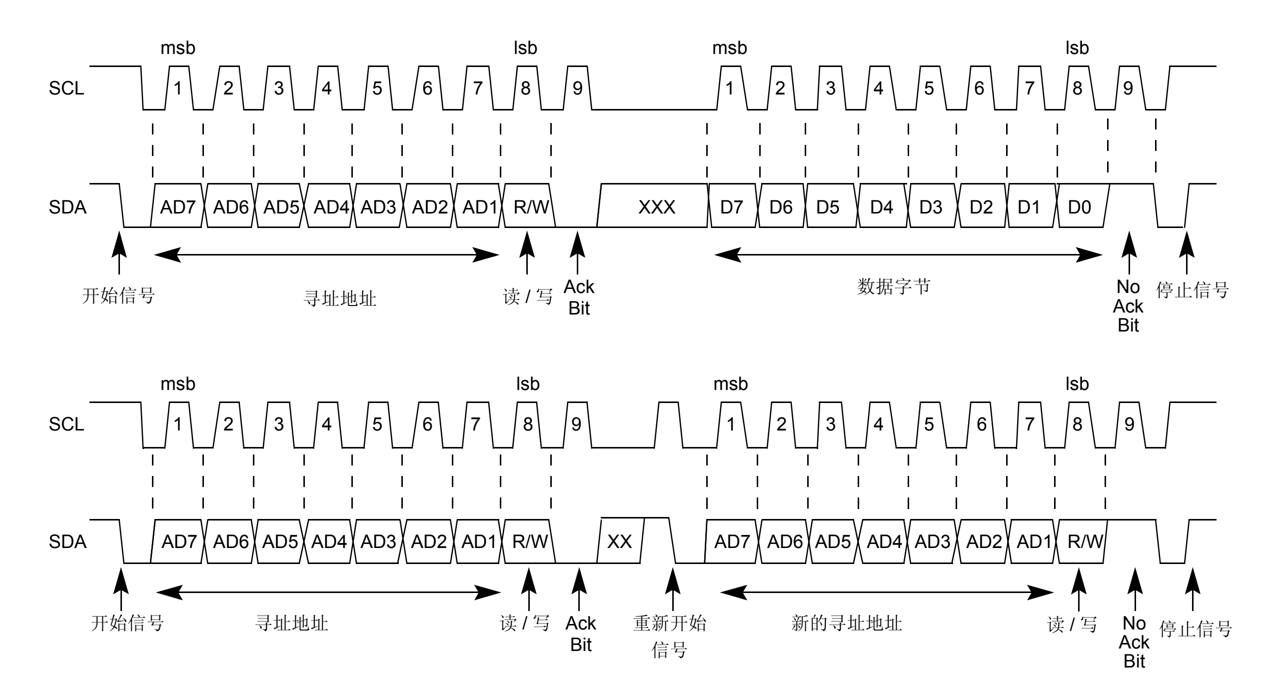


图 11-9. IIC 总线发送信号

11.4.1.1 开始信号

当总线空闲时，无主机设备使用总线 （SCL 和 SDA 逻辑高），主机可以通过发送一个开始信号开始通信。如图 11-9 所示，开始信号定义为当 SCL 为高时， SDA 从高变为低的转变。该信号表征新数据传送的开始 （每次传送的数据可能包括几个字节），同时清除所有从机空闲状态。

11.4.1.2 从机地址传送

在开始信号之后，被传输的第一个字节是主机发送的从机地址。这是七位寻址地址，它后面跟随着 R/W 位。该 R/W 位告诉从机数据传送的方向。

1= 读发送，从机发送数据到主机。

0 = 写发送，主机发送数据到从机。

仅当主机地址和从机地址相匹配时才返回一个应答信号。这通过在第九个时钟拉低 SDA 来完成 （见图 11-9）。

系统中没有两个从机地址是相同的。如果 IIC 工作于主机方式，它绝不会发送一个与它自身从机地址相同的地址。 IIC 不能同时作为主机和从机方式使用。然而，如果在一个地址周期内仲裁丢失， IIC 将恢复到从机模式，并且在被另外一个主机寻址时，能正常操作。

11.4.1.3 数据发送

在成功地完成从机寻址之前，数据以字节为单位发送，方向由寻址主机发送的读写位 （R/W）确定。

在一个地址周期后的所有传输都可看做是数据传输，即使当中为从机携带了子地址信息。

每字节数据是 8 位。数据只有在 SCL 为低时才可能发生改变，为高时必须保持不变，见图 11-9。每一个数据位在 SCL 上有一个时钟脉冲相对应， msb 最先被发送。每个数据字节后跟随着第九 （应答）位，应答位

是由接收设备标识返回的。通过在第九个时钟拉低 SDA 标识一个应答信号。简而言之，一次完整的数据通信需要九个时钟脉冲。

如果从机接收器在第九个位时间没有应答主机，那么 SDA 数据线被从机置高电平。主机将该失败的应答解释为是一次不成功的数据传输。

如果主机接收器在一个字节数据传送后没有应答从机发送器，那么从机认为数据传输过程结束并释放 SDA 数据线。

无论哪一种情况，数据传送失败，主机做下面两件事情之一：

- 产生一个停止信号，放弃总线
- 产生一个重新开始信号开始一个新的传输

11.4.1.4 停止信号

主机可以通过产生一个停止信号结束本次数据通信以释放总线。然而，主机可能没有先产生一个停止信号，而直接产生一个紧跟着调用命令的开始信号。这被称为重新开始。停止信号被定义为当 SCL 为逻辑 1 时， SDA 由低电平转变为高电平产生的信号。（见图 11-9）

即使从机已经产生一个应答信号说明从机必须释放总线，主机也可产生一个停止信号结束数据通信。

11.4.1.5 重新开始信号

如图 11-9 所示，重新开始信号是一个没有先产生结束通信的停止信号的开始信号。它被用于主机与另一个从机或与同一从机在不同模式下相互通信而不释放总线。

11.4.1.6 仲裁程序

IIC 总线是多主机总线，它允许多个主机与它连接。在同一时间 , 如果有两个或更多主机试图取得总线控制权，由时钟同步程序决定该总线时钟，该总线时钟低电平周期等于最长的时钟低电平周期，高电平周期等于最短的时钟高电平周期。这些竞争主机的相对优先级是由一个数据仲裁程序决定。当一个总线主机发送逻辑 1 而其他主机发送逻辑 0 时，该总线主机丢失仲裁。未取得总先控制权的主机立即转为从机接收模式并停止 SDA 输出。在此情况下从主机模式到从机模式的跳变不产生停止条件。同时一个状态位通过硬件被置位表明仲裁丢失。

11.4.1.7 时钟同步

因为 SDA 数据实行线与逻辑， SCL 时钟线上的一个由高到低的跳变将影响所有连接到总线上的设备。设备开始计数其低电平周期并在一个设备时钟变低后，它将保持 SCL 时钟线为低电平直到高电平状态来到。然而，如果某一个设备时钟仍然处于低电平，那么其他设备时钟由低变高的转变可能不会改变 SCL 时钟线的状态。因此同步时钟 SCL 被保持于最长低电平周期。在这期间带有较短低电平周期的设备进入高等待状态 （见图 11-10）。当所有设备结束其低电平周期时，同步时钟 SCL 线被释放并拉高。此时，设备时钟和 SCL 时钟线状态没有区别，所有设备开始高电平周期。第一个完成高电平周期的设备再次拉低 SCL 时钟线。

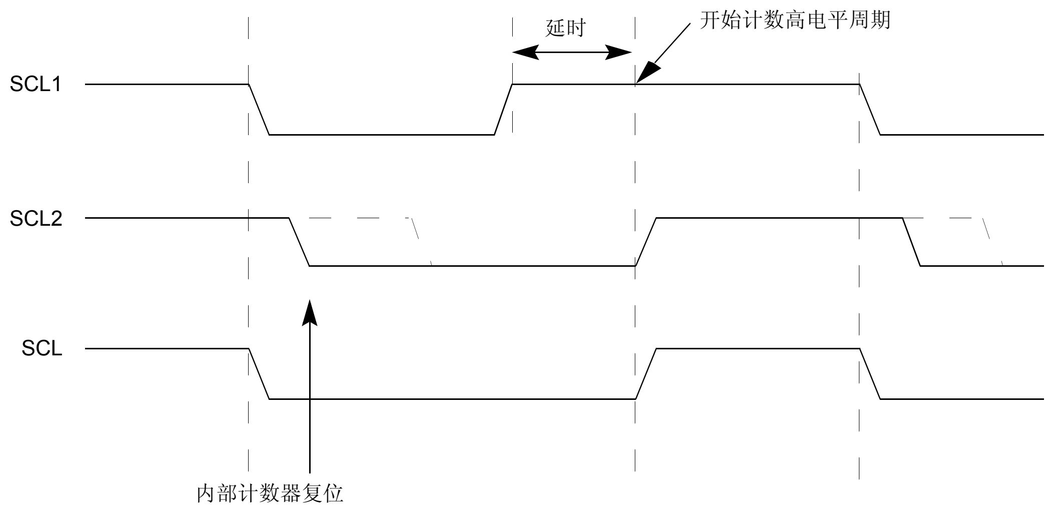


图 11-10. IIC 时钟同步

11.4.1.8 握手

时钟同步机制在数据发送中可以被用作握手机制。从设备在完成一个字节 （9 位）发送后可能保持 SCL 低电平。在这种情况下，握手信号暂停总线时钟并强制主机时钟进入等待状态直到该从机释放 SCL 时钟线。

11.4.1.9 时钟扩展

时钟同步机制可被从机用来降低发送的位速度。在主机完成驱动 SCL 时钟线为低后从机在必需的情况下可以驱动它为低电平然后释放它。如果从机时钟线的低电平周期比主机的长，那么 SCL 总线信号低电平周期被扩展。

11.4.2 10 位地址

对于 10 位寻址， 0X11110 被用于第一个地址的头 5 位。包含 10 位寻址的传送有多种读 / 写形式的组合。

11.4.2.1 主机发送从机接收

数据发送的方向不改变 （见表 11-9）。当在开始条件后发送一个 10 位地址时，每个从机将该地址的头七位与自己的地址相比并测试第八位 （读写方向位）是否为 0。超过一个设备能够匹配该地址并返回一个应答信号 （A1）。然后，这些匹配的从机用自身地址与第二个字节的八位相比较。仅有一个从机匹配，并产生一个应答信号 （A2）。该匹配的从机保持被主机寻址直到它接收到一个停止信号 （P）或一个后面跟随着不同从机地址的重新开始信号 （Sr）。

表 11-9. 10 位地址主机发送从机接收

S	从机地址的头 7 位 11110+AD10+AD9	R/W 0	A1	从机地址第二个字节 AD[8:1]	A2	数据	A	…	数据	A/A	P

在主机发送器发送 10 位地址的第一个字节，从机接收器会看到 IIC 中断。必须通过软件来确保 IICD 的内容被忽略且不被作为这次中断的有效数据。

11.4.2.2 主机接收从机发送

传送方向在第二个读写位后被改变 （见表 11-10）。直到并包括应答位 A2，程序与主机发送从机接收方式相同。在重新开始信号后 （Sr），一个匹配的从机记住它已被寻址过。然后，从机检测 Sr 后从机地址的第一个字节的头七位是否与他们在收到开始信号 S 后的相同，然后测试第八位 （读写位）是否为 1。如果存在匹配，从机认为它已被作为发送方被寻址并产生应答信号 A3。该从机发送器保持被寻址直到它接收到停止信号 （P）或后面跟着不同从机地址的重新开始信号 （Sr）。

在重新开始信号之后 （Sr），所有其它的从设备也用自身地址与该从机地址第一个字节的头七位相比较并测试第八位 （读写位）。如果读写位为 1 （10 位地址设备）或从机地址 11110XX （7 位地址设备）未匹配的话，他们不能被寻址。

表 11-10. 10 位地址主机接收从机发送

S	从机地址的头 7 位 11110+AD10+AD9	R/W 0	A1	从机地址第二个字节 AD[8:1]	A2	Sr	从机地址头 7 位 11110+AD10+AD9	R/W 1	A3	数据	A	…	数据	A/A	P

在主机接收器发送 10 位地址的第一个字节后，从机发送器看到一个 IIC 中断。必须通过软件确保 IICD 的内容被忽略且不被当作这次中断的有效数据。

11.4.3 一般寻址

被请求的一般寻址可以是 7 位或 10 位地址。如果 GCAEN 位被置位， IIC 匹配一般寻址和匹配它自身的从机地址的方式相同。当 IIC 响应一般寻址时，它可以作为从机接收器且在该地址周期后 IAAS 位被置位。在第一个字节传送后，软件必须读取 IICD 寄存器来决定匹配的地址是其自身的从机地址还是一般寻址。如果该值是 00，该匹配是一般寻址。如果 GCAEN 位被清零， IIC 通过不发送应答信号来忽略任何来自一般寻址数据。

11.5 复位

IIC 在复位后被禁止。 IIC 不能触发 MCU 复位。

11.6 中断

IIC 产生单一中断。

若 IICIE 位被置位，当表 11-11 中列举的任何一个事件发生时，将产生 IIC 中断。中断由 IICIF 位 （位于 IIC 状态寄存器）驱动，用 IICIE 位 （位于 IIC 控制寄存器）屏蔽。在终端服务例程中 IICIF 位必须通过软件方法写一个 1 给它来清零。通过读状态寄存器可确定中断类型。

表 11-11. 中断摘要

中断源	状态	标志	使能
完成一个字节传送	TCF	IICIF	IICIE
标准寻址匹配	IAAS	IICIF	IICIE
仲裁丢失	ARBL	IICIF	IICIE

11.6.1 字节传送中断

TCF 位 （传送完成标志）在第九个时钟的下降沿被置位以表明一个字节传送完成。

11.6.2 地址检测中断

当寻址地址和预设的从机地址 （IIC 地址寄存器）匹配或 GCAEN 位被置位并且接收到一般寻址时，状态寄存器中的 IAAS 位被置位。若 IICIE 被置位，则 CPU 被中断。 CPU 检测 SRW 位并据此设置发送模式。

11.6.3 仲裁丢失中断

IIC 是一个多主机总线，允许多个主机与它连接。在同一时间，如果有两个或更多的主机试图取得总线控制权，这些竞争主机的相对优先级由一个数据仲裁程序决定。当它丢失该数据仲裁程序且状态寄存器中的 ARBL 位被置位时， IIC 模块确认中断。

在下列情形下仲裁丢失：

- 在一个地址或数据传送周期内主机为高电平时， SDA 数据线采样为低电平。
- 在一个数据接收周期的应答位内主机为高电平时， SDA 数据线采样为低电平。
- 当总线忙时试图开始一个周期。
- 在从机模式下请求重新开始周期。
- 当主机没有请求停止信号时检测到停止信号。

该位必须通过软件方法写 1 来清零。

11.7 初始化 / 应用信息

模块初始化 （从机）

1. 写 IICC2
   - 禁止或允许一般寻址
   - 选择 10 位或 7 位寻址模式
2. 写 IICA
   - 设置从机地址
3. 写 IICC1
   - 使能 IIC 和中断
4. 为传送数据初始化 RAM 变量 （IICEN=1， IICIE=1）
5. 初始化 RAM 变量用来完成图 11-12 中所示的例程

模块初始化 （主机）

1. 写：IICF
   - 设置 IIC 波特率
2. 写 IICC1
   - 使能 IIC 和中断
3. 为传送数据初始化 RAM 变量 （IICEN=1， IICIE=1）
4. 初始化 RAM 变量用来完成图 11-12 中所示的例程
5. 写 IICC1
   - 使能 TX
6. 写：IICC1
   - 使能 MST （主机模式）
7. 写：IICD
   - 目标从机的地址 （这个字节的 lsb 位决定通信为主机接收还是发送）

模块使用

图 11-12 所示的例程可以处理主机和从机 IIC 操作。对于从机操作来说，一个包含适当地址的 IIC 消息进入，则开始 IIC 数据通信。对于主机操作来说，必须通过写 IICD 寄存器来开始通信。

寄存器模型

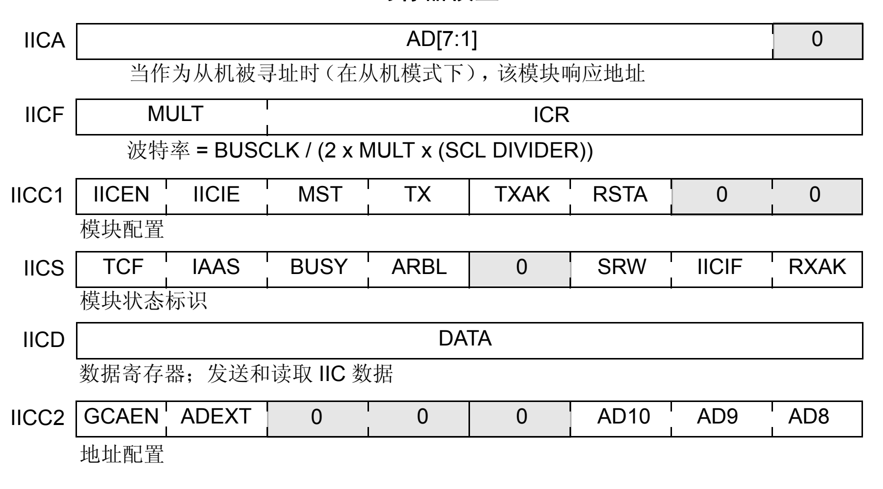
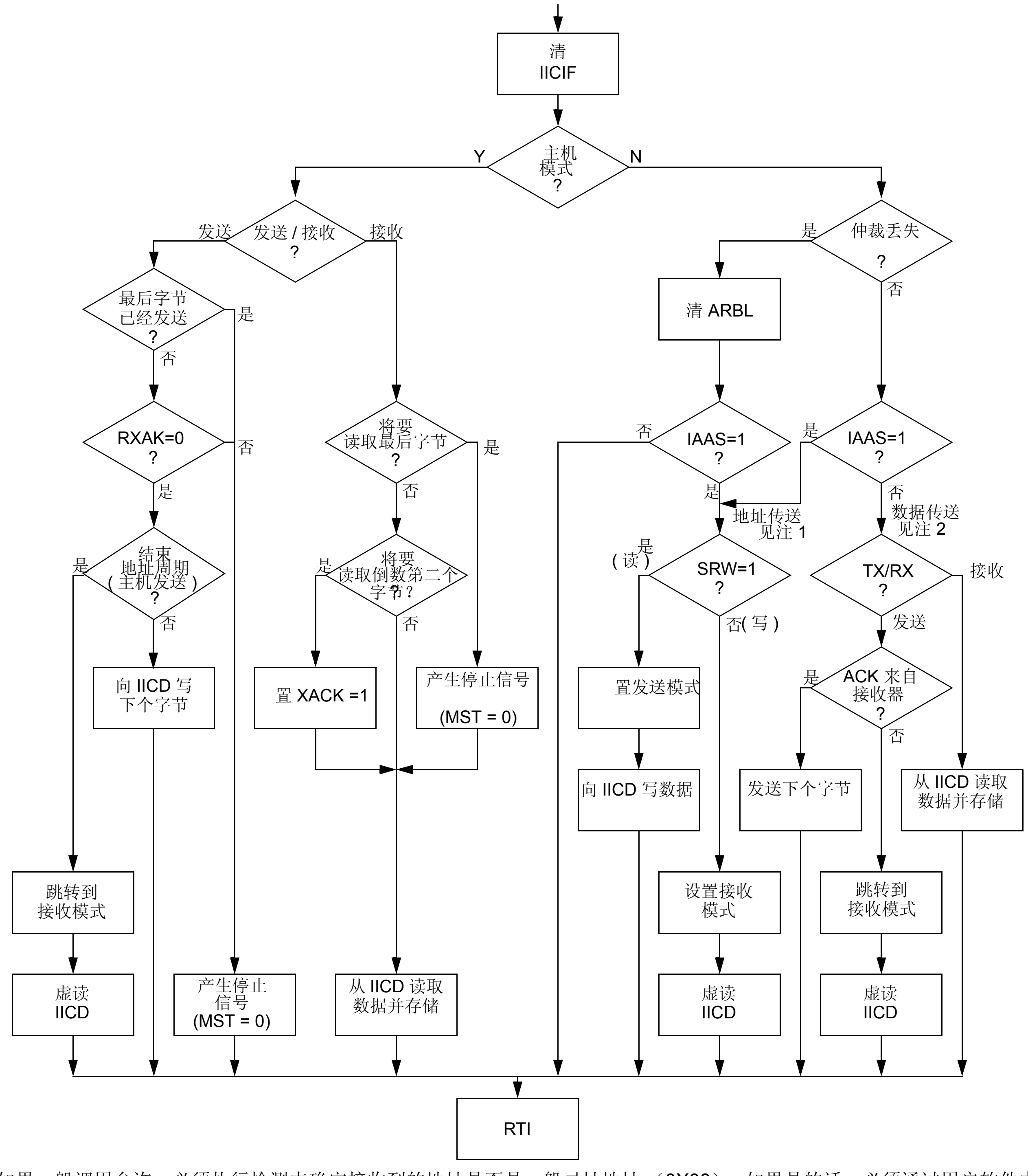


图 11-11. IIC 模块快速参考



注

1 如果一般调用允许，必须执行检测来确定接收到的地址是否是一般寻址地址 （0X00）。如果是的话，必须通过用户软件来处理它。

2 当 10 位地址被用来寻址从机，从机看到中断及其后扩展地址的第一个字节。对此中断，用户软件必须确保 IICD 的内容被忽略且不作为一个有效数据传送。

图 11-12. 典型的 IIC 中断流程

第 12 章
键盘中断 (S08KBIV1)

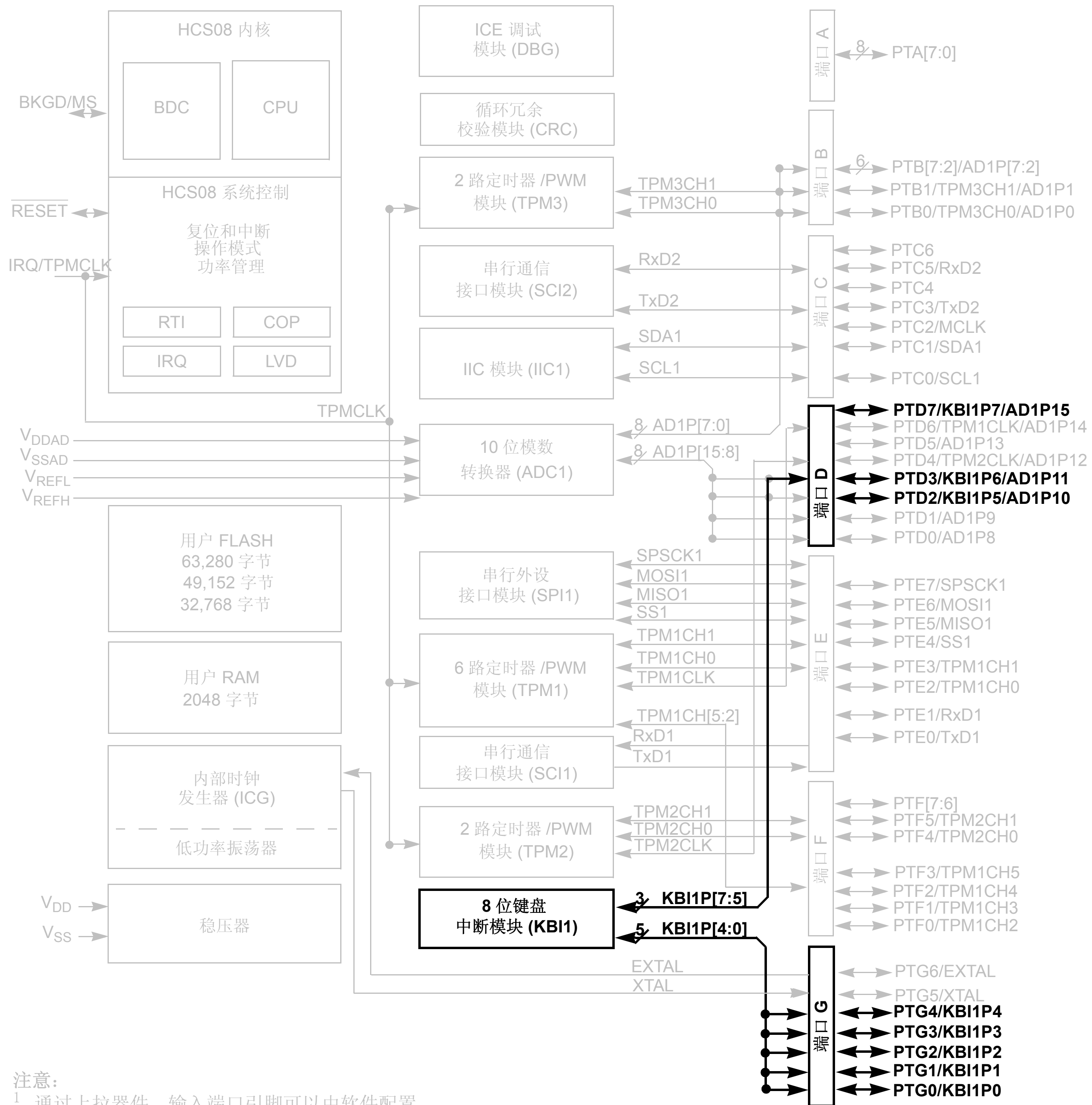
12.1 介绍

MC9S08AC60 系列 包含一个最多可达 8 个键盘中断输入的 KBI 模块。键盘中断输入的数量视不同封装而定。

12.1.1 特性

键盘中断 （KBI）模块的特性包括：

- 四个下降沿 / 低电平敏感
- 四个下降沿 / 低电平敏感或上升沿 / 高电平敏感
- 可以选择仅边沿敏感或边沿兼电平敏感
- 通用中断标记和中断使能控制
- 能够从停止 3 或等待模式唤醒微控制器



注意：

[1] 通过上拉器件，输入端口引脚可以由软件配置。

[2] 如果 IRQ 使能 (IRQPE = 1)，引脚包含软件可配置的上拉 / 下拉器件。如果选择上升边沿检测，则下拉使能 (IRQEDG = 1)。

[3] 引脚包含集成上拉器件。

[4] PTD3、PTD2、PTD7 和 PTG4 既包含上拉也包含下拉器件。当 KBI 使能时 (KBIPEn = 1)，且选择了上升边沿 (KBEDGn = 1)，则下拉使能 。

[5] TPMCLK、 TPM1CLK 和 TPM2CLK 选项可通过软件配置； 复位后， TPM1CLK、 TPM2CLK 和 TPMCLK 分别提供给 TPM1、 TPM2 和 TPM3。

图 12-1. 显亮 KBI 模块的 MC9S08AC60 系列模块图

12.1.2 KBI 模块结构图

图 12-2 为 KBI 模块的结构图。

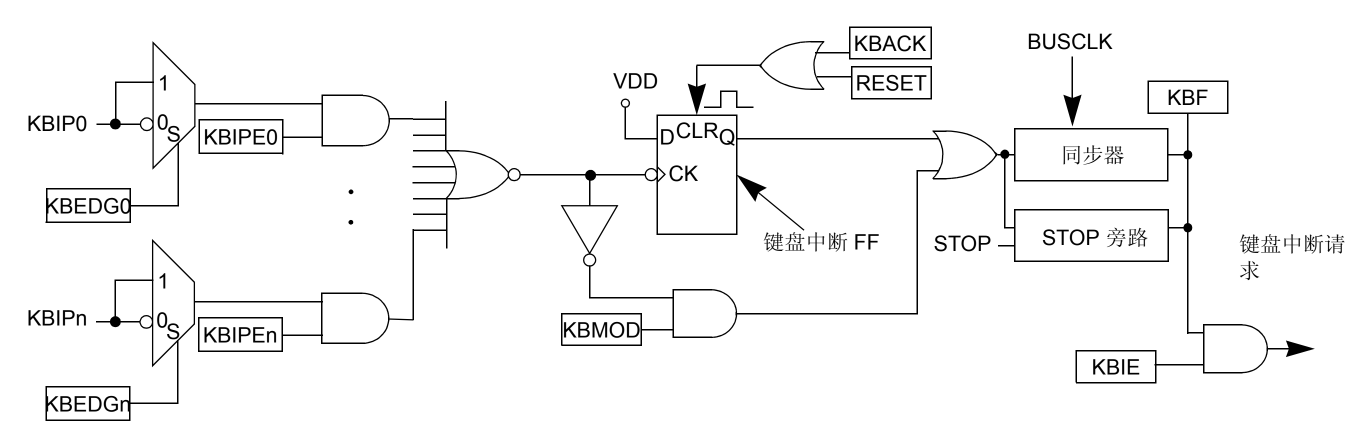


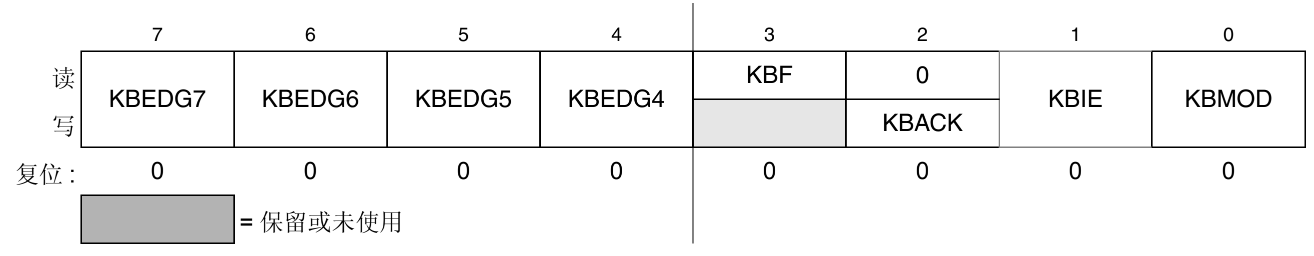
图 12-2. KBI 模块结构图

12.2 寄存器定义

本节描述 KBI 模块所有的寄存器及其相关的控制位。

KBI 寄存器的绝对地址参见本手册内存一章中的直接页寄存器的概述。这里通过名称来引用这些寄存器和控制位。 Freescale 提供一个头文件把它们的名称翻译为绝对地址。

12.2.1 KBI 状态控制寄存器 （KBISC）



	7	6	5	4	3	2	1	0
读	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBF	0	KBIE	KBMOD
写						KBACK		
复位 :	0	0	0	0	0	0	0	0

= 保留或未使用

图 12-3. KBI 状态控制寄存器 （KBISC）

表 12-1. KBISC 寄存器域描述

域	描述
7:4 KBEDG[7:4]	**KBI 端口边沿选择位** —— 当引脚被设置为 KBI 中断输入 （KBIPEn = 1）时，每个可读写位设置相应 KBI 端口引脚的触发事件需要的边沿和 / 或电平的极性。参见 KBIMOD 控制位，它决定引脚是边沿触发还是边沿和电平触发。 0 下降沿 / 低电平。 1 上升沿 / 高电平。
3 KBF	**键盘中断标识** —— 只要在使能的 KBI 引脚上检测到选定的边沿事件，这个只读位就被设置。通过写 1 到 KBACK 控制位，可清零本标识位。如果 KBIMOD = 1 选择了边沿和电平操作而且使能的 KBI 引脚保持引起事件需要的电平，标识位保持置位 1。 KBF 可以用作软件轮询标识 （KBIE = 0）或向 CPU 产生硬件中断请求 （KBIE = 1）。 0 无 KBI 中断挂起。 1 有 KBI 中断挂起。

表 12-1. KBISC 寄存器域描述 (续)

域	描述
2 KBACK	键盘中断应答 —— 通过置 1 KBACK，本只写位 （读为 0）被用来清 KBF 状态标识。如果 KBIMOD=1 选择了边沿和电平操作而且使能的 KBI 引脚保持引起事件需要的电平， KBF 保持置位 1，所以置 1 KBACK 不会清 KBF 标识。
1 KBIE	键盘中断使能——当 KBF 状态标识位为 1 时，这个可读写控制位决定是否产生硬件中断。当 KBIE = 0 时，没有硬件中断产生，但是 KBF 仍可作为软件轮询使用。 0 KBF 不产生硬件中断 （使用轮询）。 1 当 KBF=1 时，发出 KBI 硬件中断请求。
0 KBIMOD	键盘检测模式——这个可读写控制位决定采用边沿检测还是边沿电平检测。 KBI 口的 0-3 位检测仅下降沿或下降沿和低电平。 KBI 口的 4-7 位可设置为检测下面中的一种： t 仅上升沿或上升沿和高电平 （KBEDGn = 1） t 仅下降沿或下降沿和低电平 （KBEDGn = 0） 0 仅检测边沿。 1 同时检测边沿和电平。

12.2.2 KBI 引脚使能寄存器 （KBIPE）

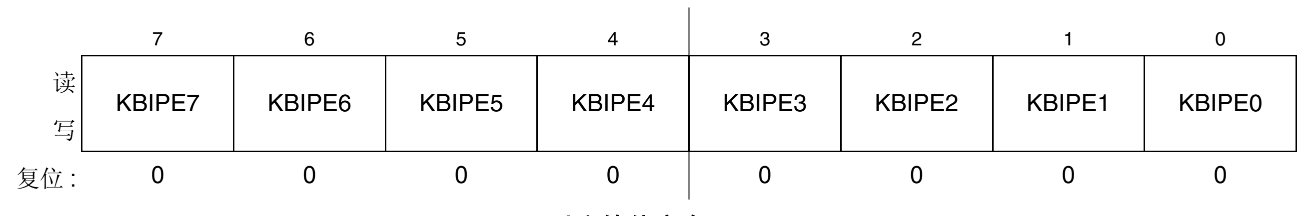


图 12-4. KBI 引脚使能寄存器 （KBIPE）

表 12-2. KBIPE 寄存器域描述

域	描述
7:0 KBIPE[7:0]	KBI 端口引脚使能位——每个可读写位决定相应的 KBI 口引脚作为键盘中断输入还是通用 I/O 引脚。 0 KBI 口的位 n 是与 KBI 无关的通用 I/O 引脚。 1 KBI 口的位 n 作为键盘中断输入。

12.3 功能描述

12.3.1 引脚使能

KBIPE 寄存器的 KBIPEn 控制位允许用户使能 （KBIPEn = 1）所有接到 KBI 模块的与 KBI 相关的引脚的组合。 KBIPE 中为 0 的引脚作为通用 I/O 引脚，与 KBI 模块无关。

12.3.2 边沿和电平触发

同步逻辑用于检测边沿。在检测边沿前， KBI 模块中使能的键盘输入必须是非触发需要的逻辑电平。

使能的键盘输入信号在一个总线周期内是逻辑 1，在下一个总线周期为逻辑 0，则下降沿被检测到。

键盘输入信号在一个总线周期内是逻辑 0，在下一个总线周期为逻辑 1，则上升沿被检测到。

KBIMOD 控制位可以通过置位来重新配置检测逻辑来检测边沿和电平。在 KBIMOD = 1 时，当检测到边沿时 （当一个或多个使能引脚从 0 变成 1，同时其他使能引脚仍然是 0）， KBF 状态标识置位。但是只要有使

能的引脚为 1，标识位保持置位 （不能清零）。当 MCU 进入停止模式时，由于时钟被停止，同步边沿检测逻辑被旁路。在停止模式时， KBI 输入作为异步的电平触发输入来唤醒 MCU。

12.3.3 KBI 中断控制

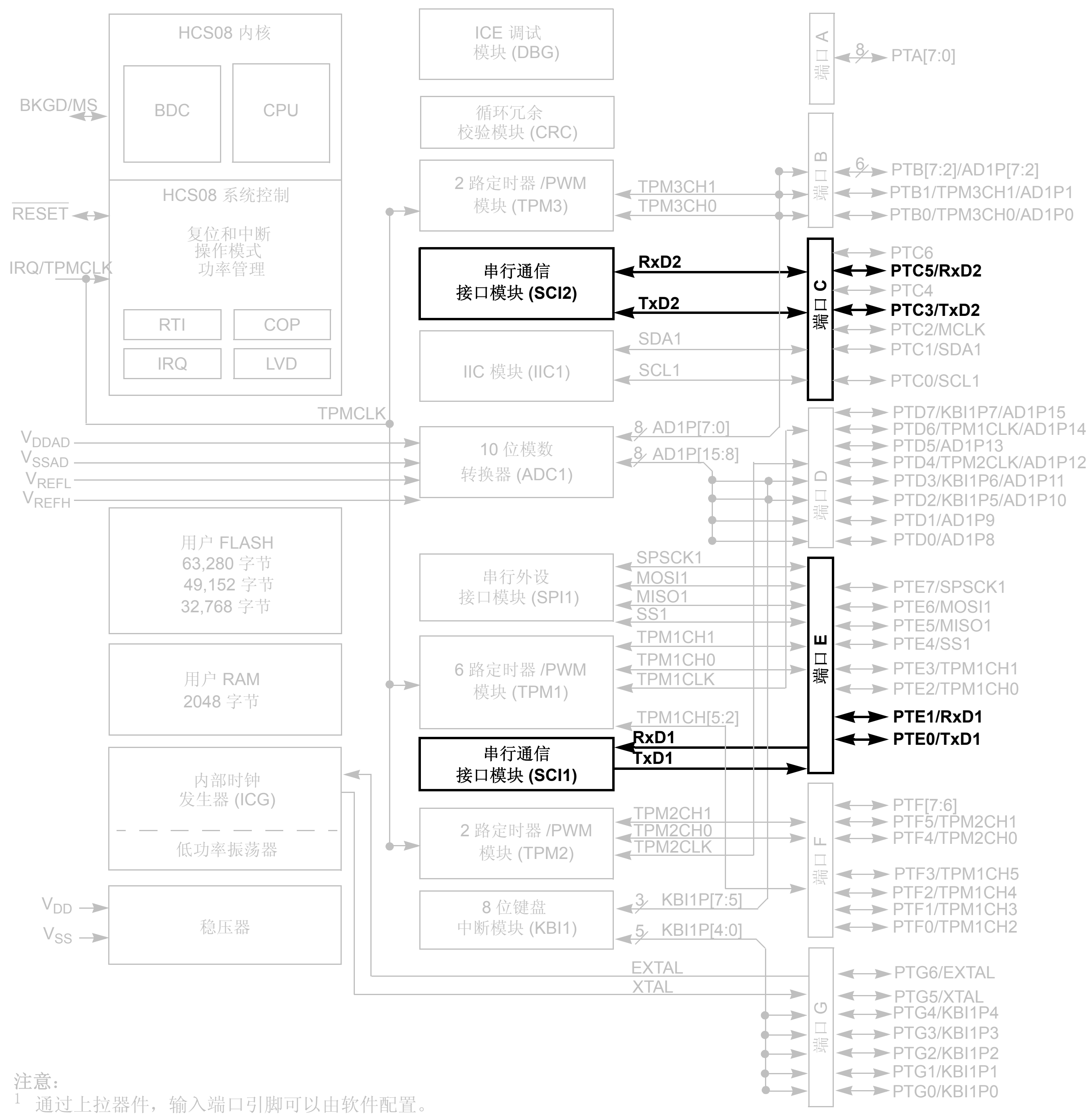
当 KBI 输入引脚检测到边沿事件， KBF 状态标识置 1。如果 KBISC 寄存器的 KBIE = 1，当 KBF=1 时，发出硬件中断请求。通过置 1 键盘应答位 （KBACK），对 KBF 标识清零。

当 KBIMOD = 0 （选择仅边沿操作）时，通过对 KBACK 置 1 来对 KBF 清零。当 KBIMOD =1 （选择边沿和电平操作）时，只要任何键盘输入为 1， KBF 就不能被清零。

第 13 章
串行通信接口 (S08SCIV4)

13.1 介绍

根据封装情况，MC9S08AC60 系列最多提供两个独立的串行通信接口（SCI) 模块。SCI 有时也被称作通用异步收发器 （UART) 。在 MC9S08AC60 系列中不支持停上 1 模式，阅读本章时请忽略这些内容。



注意：

1 通过上拉器件，输入端口引脚可以由软件配置。

2 如果 IRQ 使能 (IRQPE = 1)，引脚包含软件可配置的上拉 / 下拉器件。如果选择上升边沿检测，则下拉使能 (IRQEDG = 1)。

3 引脚包含集成上拉器件。

4 PTD3、PTD2、PTD7 和 PTG4 既包含上拉也包含下拉器件。当 KBI 使能时 (KBIPEn = 1)，且选择了上升边沿 (KBEDGn = 1)，则下拉使能 。

5 TPMCLK、 TPM1CLK 和 TPM2CLK 选项可通过软件配置； 复位后， TPM1CLK、 TPM2CLK 和 TPMCLK 分别提供给 TPM1、 TPM2 和 TPM3。

图 13-1. 显亮 SCI 模块及其引脚的 MC9S08AC60 系列模块图

13.1.1 特性

SCI 模块具有以下特性：

- 全双工，标准的不归零 （NRZ）格式。
- 发送器和接收器内具有各自的双缓冲，并可分别使能。
- 可编程的波特率 （13 位模因子）。
- 中断驱动或轮询操作：
  - 发送数据寄存器空和发送完成
  - 接收数据寄存器满
  - 接收溢出，奇偶校验错误，帧错误，和干扰错误
  - 接收器空闲检测
  - 接收引脚上的有效边沿
  - 间断检测支持 LIN
- 硬件奇偶产生和校验。
- 可编程的 8 位或 9 位的字符长度。
- 空闲线或地址标记唤醒接收器。
- 可选的 13 位间断字符产生或 11 位的间断字符检测。
- 可选的发送器输出极性。

13.1.2 操作模式

在下列模式中 SCI 操作的细节，参见第 13.3 节 功能描述。

- 8 位或 9 位的数据模式
- 停止模式操作
- 轮询模式
- 单线模式

13.1.3 框图

图 13-2 为 SCI 的发送器部分。

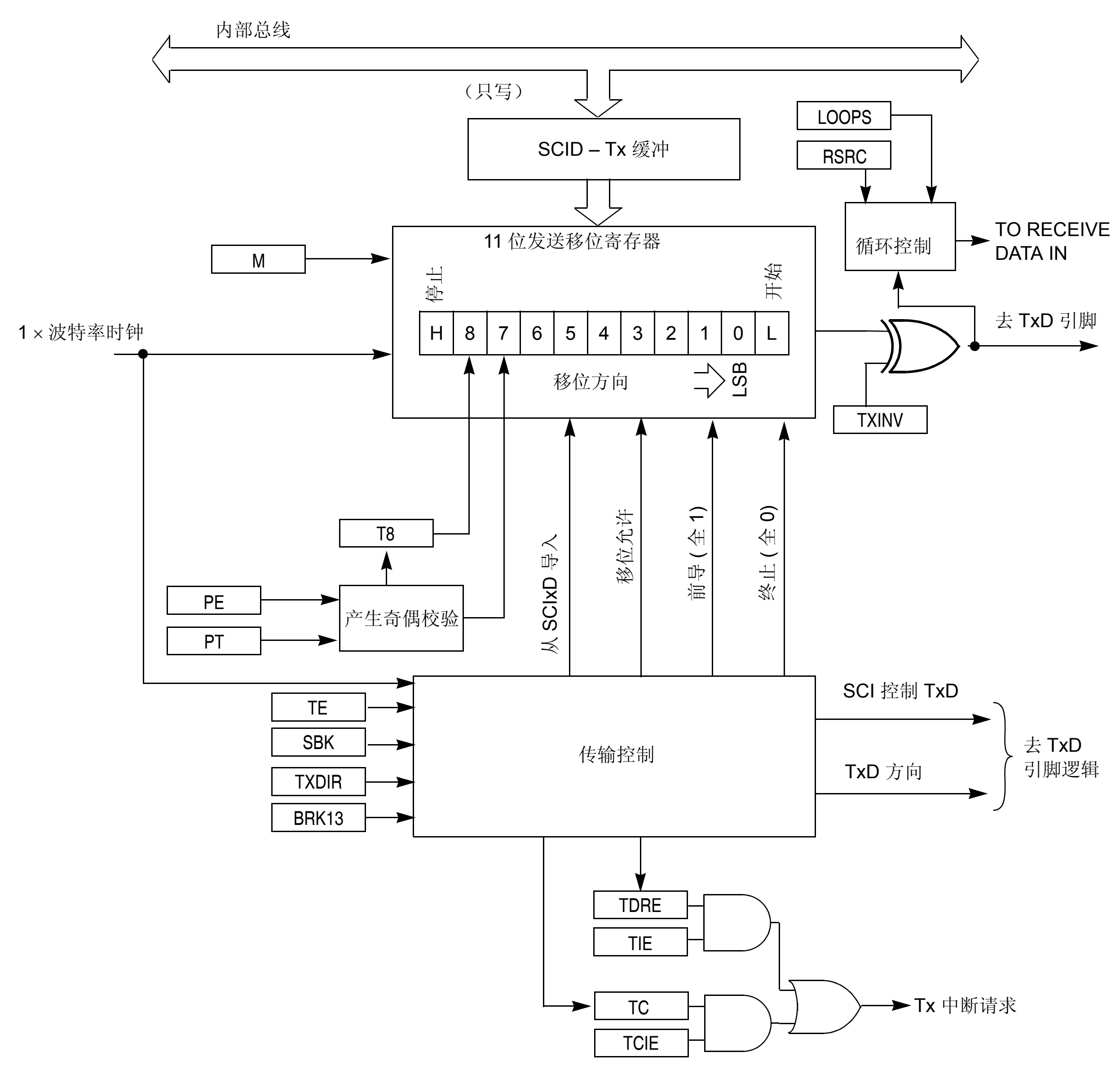


图 13-2. SCI 发送器原理结构图

图 13-3 为 SCI 的接收器部分。

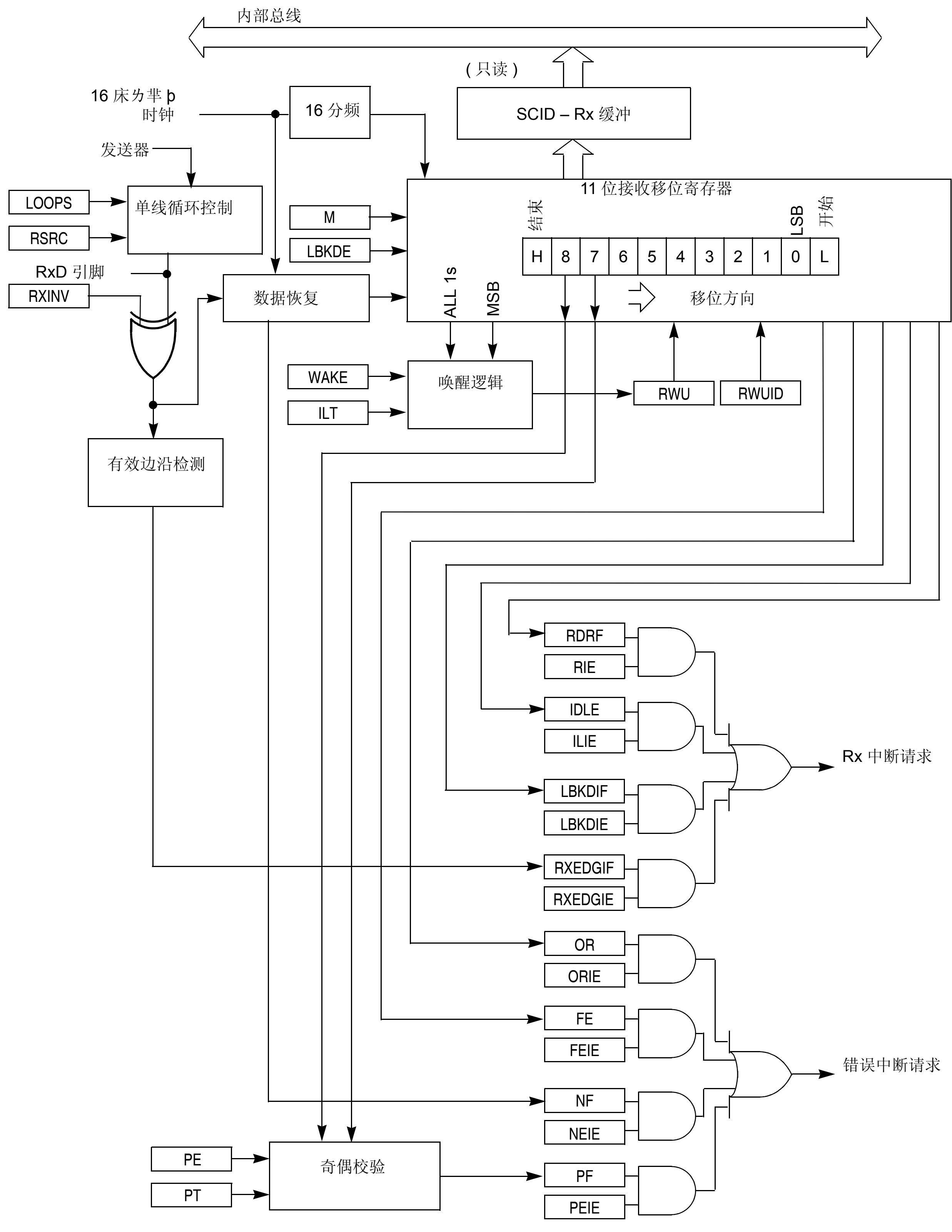


图 13-3. SCI 接收器原理结构图

13.2 寄存器定义

SCI 有 8 个 8 位寄存器，分别用于控制波特率、选择 SCI 选项、报告 SCI 状态和发送 / 接收数据。

SCI 寄存器的绝对地址参见本数据手册内存一章中的直接页寄存器的概述。这里通过这些寄存器和控制位的名称来引用它们。通常， Freescale 提供一个头文件把它们的名称翻译为绝对地址。

13.2.1 SCI 波特率寄存器 （SCIxBDH、 SCIxBDL）

这两个寄存器控制产生 SCI 波特率的预分频因子。设置 13 位的波特率 [SBR12:SBR0]，先写新值的高半部分到 SCIxBDH，然后写 SCIxBDL。直到写完 SCIxBDL， SCIxBDH 的值，波特率才改变。

SICxBDL 复位后的值非零，所以复位后波特率产生器仍然禁止，直到接收或发送第一次被允许（SCIxC2 的 RE 或 TE 置 1）。

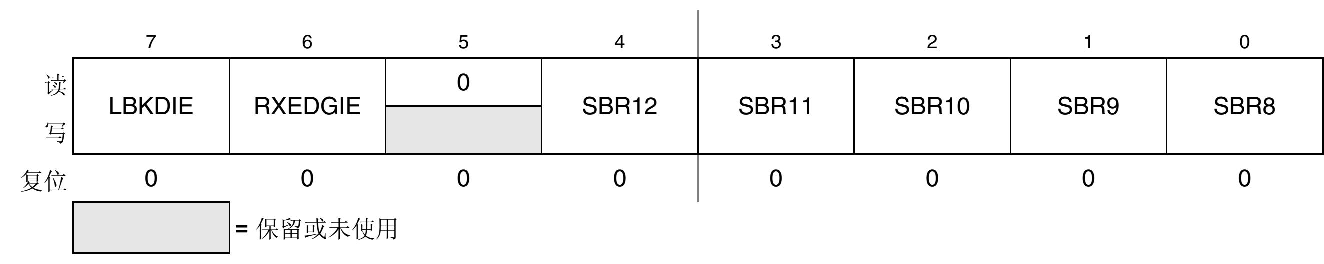


图 13-4. SCI 波特率寄存器 （SCIxBDH）

表 13-1. SCIxBDH 寄存器域描述

域	描述
7 LBKDIE	LIN 中止检测中断使能 （对于 LBKDIF） 0 禁止来自 LBKDIF 的硬件中断 （使用轮询）。 1 当 LBKDIF 标志为 1 时，发送硬件中断请求。
6 RXEDGIE	RxD 输入有效边中断使能 （对 RXEDGIF） 0 禁止来自 RXEDGIF 的硬件中断 （使用轮询）。 1 当 RXEDGIF 标志为 1 时，发送硬件中断请求。
4:0 SBR[12:8]	波特率分配因子——13 位的 SBR[12:0] 记作 BR，可以设置 SCI 波特率产生器的分频率。当 BR=0 时， SCI 波特率产生器被禁止以减少电路消耗。 BR 可以从 1 到 8191， SCI 波特率 =BUSCLK/(16*BR)。其他的 BR 位请参见表 13-2。

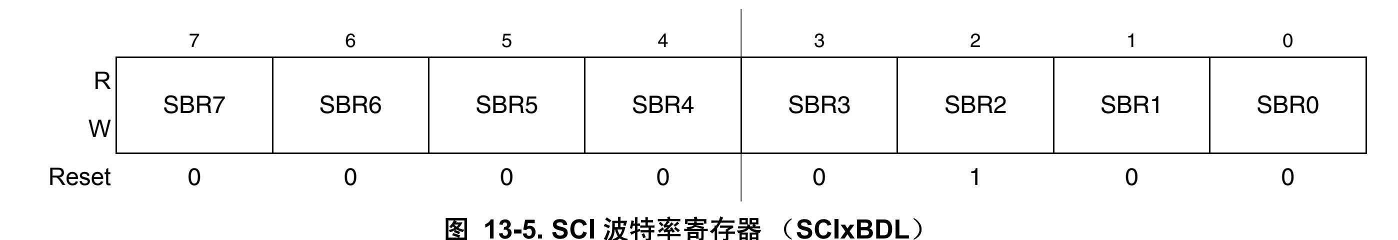


图 13-5. SCI 波特率寄存器 （SCIxBDL）

表 13-2. SCIxBDL 寄存器域描述

域	描述
7:0 SBR[7:0]	波特率分配因子——13 位的 SBR[12:0] 记作 BR，可以设置 SCI 波特率产生器的分频率。当 BR=0 时， SCI 波特率产生器被禁止以减少电路消耗。 BR 可以从 1 到 8191， SCI 波特率 =BUSCLK/(16*BR)。其他的 BR 位请参考表 13-1。

13.2.2 SCI 控制寄存器 1 （SCIxC1）

可读写寄存器，用于控制多个 SCI 的可选功能。

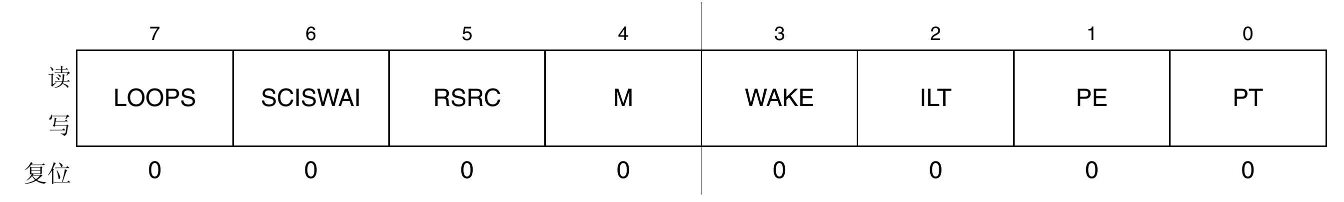


图 13-6. SCI 控制寄存器 1 （SCIxC1）

表 13-3. SCIxC1 寄存器域描述

域	描述
7 LOOPS	循环模式选择——选择循环模式和正常的全双工模式。当 LOOPS=1 时，此时发送器的输出连接到接收器的输入。 0 正常操作 --- RxD 和 TxD 采用不同的引脚。 1 循环或者单线模式，此时发送器的输出连接到接收器的输入。
6 SCISWAI	等待模式下 SCI 停止位 0 在等待模式下， SCI 时钟继续运行，所以 SCI 可以作为唤醒 CPU 的中断源。 1 在等待模式下， SCI 时钟停止运行。
5 RSRC	接收源选择——仅在 LOOPS 位为 1 时有意义。当 LOOPS=1 时，接收器输入在内部连接到 TxD 引脚上，RSRC 决定引脚是否连接到发送器输出。 0 如果 LOOPS=1， RSRC=0 时，内部循环模式和 SCI 不使用 RxD 引脚。 1 单线 SCI 模式， TxD 引脚连接到发送器输出和接收器输入。
4 M	9 位或 8 位模式选择 0 正常——开始位 +8 位数据位 （首先传输最低有效位 LSB） + 停止位。 1 接收和发送使用 9 位数据字符 开始位 +8 位数据位 （首先传输最低有效位 LSB） +9 位数据位 + 停止位。
3 WAKE	接收器唤醒方式选择——更多信息参见 13.3.3.2 节 接收唤醒。 0 空闲线唤醒。 1 地址符号唤醒。
2 ILT	空闲线类型选择——设置这个位为 1 保证字符后面的停止位和逻辑 1 的个数不超过 10 或 11 位 （空闲线检测逻辑需要的高电平）。更多信息参见 13.3.3.2.1 节 空闲线唤醒 0 空闲字符位从 “开始位”开始计数。 1 空闲字符位从 “停止位”开始计数。

表 13-3. SCIxC1 寄存器域描述 (续)

域	描述
1 PE	奇偶校验允许——允许硬件奇偶校验产生和校验。当允许奇偶校验时，数据字符 （第 8 或 9 位）的最高有效位 （MSB）作为校验位。 0 不允许奇偶校验。 1 允许奇偶校验。
0 PT	奇偶校验类型——如果奇偶校验允许 （PE=1），这个位选择奇校验或偶校验。奇校验是数据字符中 1 的总数 （包括奇偶位）是奇数。偶校验是数据字符中 1 的总数 （包括奇偶位）是偶数。 0 偶校验。 1 奇校验。

13.2.3 SCI 控制寄存器 2 （SCIxC2）

可读写寄存器，用于控制多个 SCI 的可选功能。

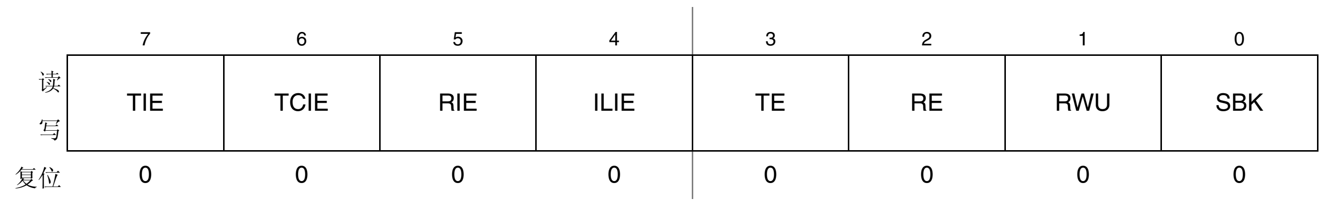


图 13-7. SCI 控制寄存器 2 （SCIxC2）

表 13-4. SCIxC2 寄存器域描述

域	描述
7 TIE	发送中断允许 （用于 TDRE） 0 TDRE 的硬件中断禁止 （使用轮询）。 1 当 TDRE 标志位是 1，发送硬件中断请求。
6 TCIE	发送完成中断允许 （用于 TC） 0 TC 的硬件中断禁止 （使用轮询）。 1 当 TC 标志位是 1，发送硬件中断请求。
5 RIE	接收中断允许 （用于 RDRE） 0 RDRE 的硬件中断禁止 （使用轮询）。 1 当 RDRE 标志位是 1，发送硬件中断请求。
4 ILIE	空闲线中断允许 （用于 IDLE） 0 IDLE 的硬件中断禁止 （使用轮询）。 1 当 IDLE 标志位是 1，发送硬件中断请求。
3 TE	发送允许 0 发送器关。 1 发送器开。 使用 SCI 发送器则 TE 要置 1，当 TE=1 时， SCI 强制 TxD 引脚作为 SCI 系统的输出。 当 SCI 设置位单线操作 （LOOPS=RSRC=1）， TXDIR 控制单个 SCI 传输线 （TxD 引脚）的传输方向。 TE=0 也可以表示一个空闲字符在排队， TE=1 则传输在处理中。更多细节参见 13.3.2.1 节 发送间隔和等待空闲。 当 TE=0 时，在引脚变回通用 I/O 引脚前，发送器一直控制 TxD 引脚，直到有数据，等待空闲或等待间隔符排队完成传输。

表 13-4. SCIxC2 寄存器域描述

域	描述
2 RE	接收允许——当接收器关时， RxD 引脚变回通用 I/O 引脚。如果 LOOPS=1，即使 RE 等于 1RxD 也会变回通用 I/O 引脚。 0 接收关。 1 接收开。
1 RWU	接收唤醒控制——该位置 1， SCI 接收器进入待命状态，等待扫描到唤醒条件。唤醒条件是信息之间的空闲线（WAKE=0，空闲线唤醒）或者字符的最高有效位是逻辑 1 （WAKE=1，地址符号唤醒）。应用软件置位 RWU，（通常）硬件唤醒条件自动清零 RWU。更多信息参见 13.3.3.2 节 接收唤醒。 0 正常 SCI 接收操作。 1 待命的 SCI 接收器等待唤醒条件。
0 SBK	发送终止——向 SBK 写一个 1 然后一个 0，一个间隔符插入传输数据流。只要 SBK=1、 10 或 11 （13 或 14）个逻辑 0 的间隔符也会插入传输数据流。第二个间隔符可能在软件清 SBK 前产生，这依赖于 SBK 在正传输的信息时复位和清零的时间选择。更多信息参见 13.3.2.1 节 发送间隔和等待空闲。 0 正常发送操作。 1 发送对了终止字符。

13.2.4 SCI 状态寄存器 1 （SCIxS1）

该寄存器有 8 个只读状态标志。写无影响。专门的软件序列 （不是写该寄存器）用来清零这些状态位。

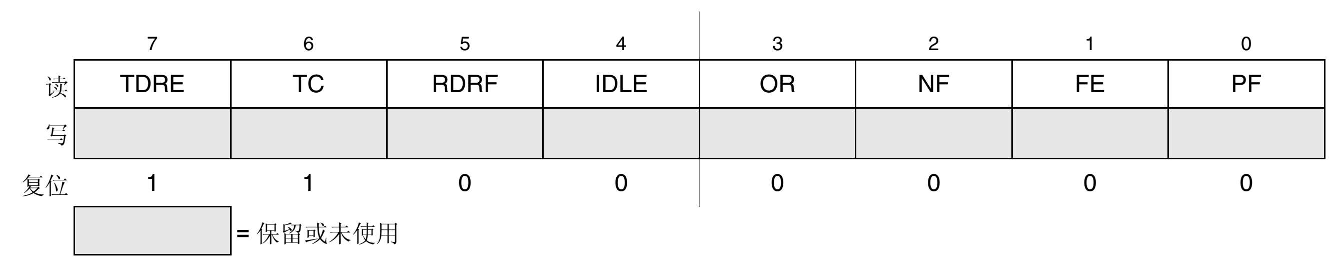


图 13-8. SCI 状态寄存器 1 （SCIxS1）

表 13-5. SCIxS1 域描述

域	描述
7 TDRE	发送缓冲区空标志——TDRE 在复位时置位。而且当发送数据从发送数据缓冲区传到发送移位寄存器为新字符留出空间时， TDRE 也置位。 SCIxS1 的 TDRE=1 然后写 SCI 数据寄存器可以清零 TDRE。 0 发送数据寄存器 （缓冲区）满。 1 发送数据寄存器 （缓冲区）空。
6 TC	发送完成标志——TC 在复位时置位。而且当 TDRE=1 并且没有数据，前导符或间隔符发送时， TC 也置位。 0 发送器忙 （发送数据，前导符或间隔符）。 1 发送器空闲。 读 SCIxS1 的 TC=1 然后做下面的 3 件事情之一， TC 就会自动清零。 t 写 SCI 数据寄存器 （SCIxD）发送新数据。 t 改变 TE 从 0 到 1 插入前导符。 t 置 1SCIxC2 的 SBK 插入间隔符。
5 RDRF	接收器满标志——当字符从接受移位寄存器传输到接收数据寄存器 （SCIxD）时， RDRF 置位。读 SCIxS1 的 RDRF=1 然后读 SCI 数据寄存器 （SCIxD）可以清 RDRF。 0 接收数据寄存器空。 1 接收数据寄存器满。

表 13-5. SCIxS1 域描述 (续)

域	描述
4 IDLE	接收器空闲标志——SCI 工作一段时间后，如果接收线空闲， IDLE 则置位。当 ILT=0 时，接收器开始计数开始位后的空闲时间。所以如果接收字符全 1，这些位的时间和停止位的时间达到了一个字符的逻辑高 （10 或 11 位依赖于 M 控制位），这会导致接收器认为检测到空闲，当 ILT=1 时，接收器直到停止位后才开始数空闲时间。所以停止位和先前字符的逻辑高不会到达一个字符逻辑高的时间，从而不会被检测为空闲线。 读 SCIxS1 的 IDLE=1 然后读 SCI 数据寄存器 （SCIxD）可以清零 IDLE。 IDLE 清零后，直到接收到一个新的字符并且 RDRF 被置位， IDLE 才能再次置位。即使接收线长时间保持空闲， IDLE 也只会被置位一次。 0 未检测到空闲线。 1 检测到空闲线。
3 OR	接收器溢出标志——当新的字符准备传到接收数据寄存器，而前面接收的数据还没有从 SCIxD 中读走， OR 置位。在这种情况下，因为没有空间把这些数据移到 SCIxD 中，所以新的字符 （和所有的相关错误信息）丢失。读 SCIxS1 的 OR=1 然后读 SCI 数据寄存器 （SCIxD）可以清零 OR。 0 未溢出。 1 接收溢出 （新数据丢失）。
2 NF	噪音标志位——在接受器中使用先进的采样技术，开始位采样 7 次，每个数据位和停止位采样 3 次。在接收到数据时， RDRF 置位的同时，如果在帧中某个采样和其他的采样不同，则标志 NF 置位。读 SCIxS1 然后读 SCI 数据寄存器 （SCIxD）可以清零 NF。 0 未检测到噪音。 1 SCIxD 中接收到的字符检测到噪音。
1 FE	帧错误标志——接收器在停止位检测到逻辑 0 时， FE 和 RDRF 置位。这表明接收器没有对齐字符帧。读 SCIxS1 的 FE=1 然后读 SCI 数据寄存器 （SCIxD）可以清零 FE。 0 未检测到帧错误。这并不确保帧时正确的。 1 帧错误。
0 PF	奇偶错误标志——当奇偶校验使能 （PE=1）并且接收到的数据中的校验位和正确的校验位不同， PE 和 RDRF 置位。读 SCIxS1 然后读 SCI 数据寄存器 （SCIxD）可以清零 PF。 0 无奇偶校验错误。 1 有奇偶校验错误。

13.2.5 SCI 状态寄存器 2 （SCIxS2）

该寄存器有 1 个只读状态标志。

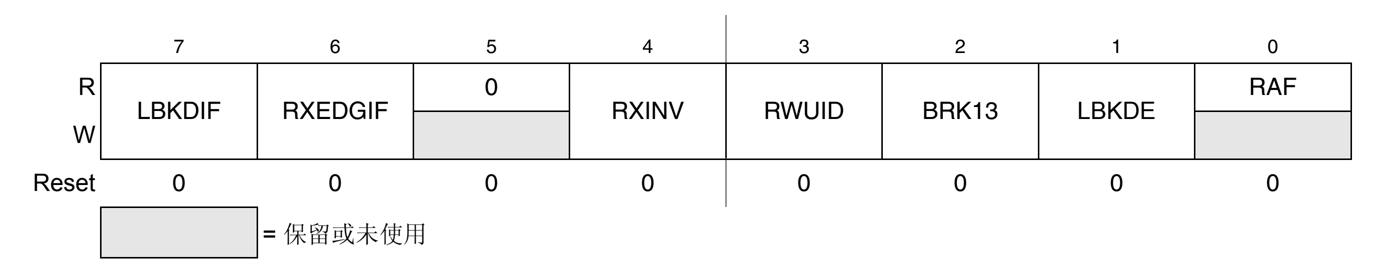


图 13-9. SCI 状态寄存器 2 （SCIxS2）

表 13-6. SCIxS2 域描述

域	描述
7 LBKDIF	LIN 终止检测中断标志——当使能 LIN 间隔检测电路，并且检测到一个 LIN 间隔符， LBKDIF 置位。向 LBKDIF 写 1， LBKDIF 清零。 0 未检测到 LIN 间隔符。 1 检测到 LIN 间隔符。
6 RXEDGIF	RxD 引脚有效边沿中断标志——当 RxD 引脚上一个有效边沿发生时 （如果 RXINV=0，下降沿，如果 RXINV=1，上升沿） RXEDGIF 置位。向其写 1， RXEDGIF 清零。 0 在接收引脚上无有效边沿发生。 1 在接收引脚上有效边沿发生。
4 RXINV[1]	接收数据反转——该位置位则接受的数据输入的极性反转。 0 接收的数据未反转。 1 接收的数据反转。
3 RWUID	接收唤醒空闲检测——RWUID 控制空闲字符是否置位 IDLE 位。 0 在接收待命状态 （RWU=1）期间，检测到空闲字符时， IDLE 不置位。 1 在接收待命状态 （RWU=1）期间，检测到空闲字符时， IDLE 置位。
2 BRK13	间隔符长度——BRK13 可以选择更长的发送间隔符长度。该位的状态不影响帧错误的检测。 0 间隔符长度是 10 字节 （M=1 则为 11）。 1 间隔符长度是 13 字节 （M=1 则为 14）。
1 LBKDE	LIN 间隔检测使能——LBKDE 可以选择更长的间隔符检测长度。当 LBKDE 置位时，可以防止帧错误 （FE）和接收数据寄存器满 （RDRF）标志置位。 0 间隔符的检测长度是 10 字节 （M=1 则为 11）。 1 间隔符的检测长度是 11 字节 （M=1 则为 12）。
0 RAF	接收器有效标志——当 SCI 接收器检测到有效位的开始， RAF 置位；当接收器检测到空闲线， RAF 清零。该状态位可以用来检查 MCU 进入停止模式前， SCI 字符是否正被接收。 0 SCI 接收器空闲等待开始位。 1 SCI 处于活动中 （RxD 输入无效）。

[1] 置位 RXINV 反转 RxD 输入的各种情况：数据位，开始位和停止位，终止和空闲。

当在 LIN 系统中使用内部振荡器时，有必要把间断检测极限值增加一位。在最坏的情况下， LIN 允许的的定时条件可能把 0x00 数据字符看做 10.26 位字节，因为在从机上运行比主机快 14%。这将触发正常的间隔检测电路 （用来检测 10 位间隔标志）。当 LBKDE 位置位时，帧错误可以减少，间隔检测阈从 10 位变成 11 位以防止把 0x00 数据字符错误地检测成 LIN 间隔符号。

13.2.6 SCI 控制寄存器 3 （SCIxC3）

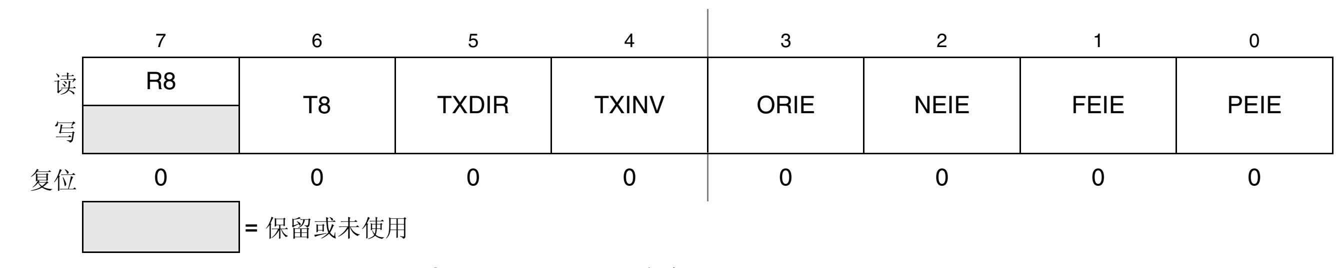


表 13-7. SCI 控制寄存器 3 （SCIxC3）

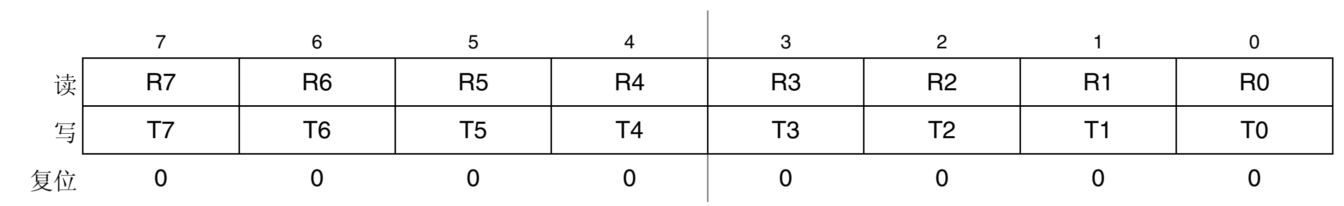
表 13-8. SCIxC3 域描述

域	描述
7 R8	接收器的第九数据位——当 SCI 设置为 9 位数据时 （M=1）， R8 可以看作 SCIxD 寄存器中数据的最高有效位左边的第九位接收数据。当读 9 位数据时，因为读 SCIxD 会自动完成清零标志 （允许在 R8 和 SCIxD 中写入新数据），在读 SCIxD 之前先读 R8。
6 T8	发送器的第九数据位——当 SCI 设置为 9 位数据时 （M=1）， T8 可以看作 SCIxD 寄存器中数据的最高有效位左边的第九位发送数据。当写 9 位数据时，在写 SCIxD 之后整个 9 位值传到 SCI 移位寄存器，所以 T8 应该在写 SCIxD 之前先写 （如果 T8 需要改变它的值）。如果 T8 不需要改变值 （比如当它用于产生标记或空间奇偶校验）则不需要每次写 SCIxD 都重写。
5 TXDIR	单线模式 TxD 引脚方向——当 SCI 被设置为单线半双工操作时 （LOOPS=RSRC=1），该位决定 TxD 引脚上的数据方向。 0 单线模式下 TxD 引脚为输入。 1 单线模式下 TxD 引脚位输出。
4 TXINV[1]	发送数据反转——该位置位则发送数据的输出的极性反转。 0 发送数据未反转。 1 发送数据反转。
3 ORIE	溢出中断使能——该位使能溢出标志 （OR），产生硬件中断请求。 0 禁止 OR 中断 （使用轮询）。 1 当 OR=1 时，发出硬件中断请求。
2 NEIE	噪音错误中断使能——该位使能溢出标志 （NF），产生硬件中断请求。 0 禁止 NF 中断 （使用轮询）。 1 当 NF=1 时，发出硬件中断请求
1 FEIE	帧错误中断使能——该位使能帧错误标志 （FE），产生硬件中断请求。 0 禁止 FE 中断 （使用轮询）。 1 当 FE=1 时，发出硬件中断请求
0 PEIE	奇偶校验错误中断使能——该位使能奇偶校验错误标志 （PF），产生硬件中断请求。 0 禁止 PF 中断 （使用轮询）。 1 当 PF=1 时，发出硬件中断请求

[1] 置位 TXINV 反转 TxD 输出的各种情况：数据位，开始位和停止位，终止和空闲。

13.2.7 SCI 数据寄存器 （SCIxD）

该寄存器实际上是两个独立的寄存器。读操作读到只读接收数据缓冲区的内容，写操作写入发送数据缓冲区。该寄存器的读和写也和 SCI 状态标志的标识自动清零机制有关。



	7	6	5	4	3	2	1	0
读	R7	R6	R5	R4	R3	R2	R1	R0
写	T7	T6	T5	T4	T3	T2	T1	T0
复位	0	0	0	0	0	0	0	0

图 13-10. SCI 数据寄存器 （SCIxD）

13.3 功能描述

SCI 是一种全双工，异步， NRZ 的串行通信。用于 MCU 和其他远程设备 （包括其他 MCU）通信。 SCI 包括一个波特率产生器，发送器和接收器。虽然使用相同的波特率产生器，发送器和接收器独立工作。正常工作时， MCU 监控 SCI 的状态，写数据发送，处理接收的数据。接下来描述每个 SCI 模块。

13.3.1 波特率产生

如图 13-11 所示， SCI 波特率产生器的时钟源时总线时钟。

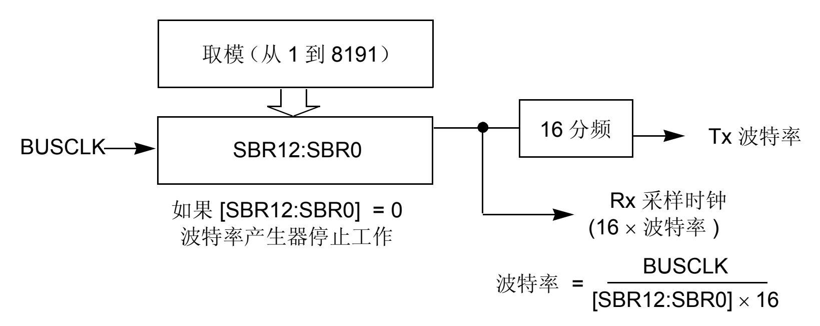


图 13-11. SCI 波特率产生

SCI 通信要求发送器和接收器 （通常从独立的时钟源获得波特率）使用相同的波特率。所允许的波特频率由两个部分决定，接收器怎样同步开始位的有效沿和位采样怎样操作。

MCU 在每个高到低转换时都会重新同步位边界。但是在最坏的情况下，在 10 或 11 位的字符帧中没有这种转换，所以在整个字符期间任何波特率的不匹配都会积累起来。在飞思卡尔半导体的 SCI 系统中，其总线频率有一个晶振得到，所允许的波特率不匹配是 4.5% （8 位数据格式）和 4% （9 位数据格式）。虽然波特率分频因子的设置并不总是产生正好匹配标准的波特率，但是通常偏差都在很小的百分比，可以保障可靠的通信。

13.3.2 发送功能描述

本节描述 SCI 接收器的框图和发送终止和空闲字符的功能。发送器的框图图 13-2 所示。

发送器输出 （TxD）空闲状态默认是逻辑高 （复位后 TXINV=0）。设置 TXINV=1，发送器输出被反转。置位 SCIxC2 的 TE，发送器使能。发送器发出一个前导符 （一个空闲状态的满字符帧）。直到发送数据缓冲区的数据准备好，发送器保持空闲。写 SCI 数据寄存器 （SCIxD）可以将数据存到发送数据缓冲区。

SCI 发送器的主要部分是发送移位寄存器， 10 位或 11 位长 （由 M 控制位决定）。在本节以后的部分，我们假定 M=0 （选择正常的 8 位数据模式）。在 8 位数据模式下，移位寄存器包含 1 个开始位， 8 个数据位，1 个停止位。当发送移位寄存器允许接受新的 SCI 字符时，等待在发送数据寄存器的值传到移位寄存器 （由波特率时钟同步）并且发送数据空 （TDRE）状态标志置位以表示 SCIxD 内的发送数据缓冲区可以写入另一个字符。

TxD 引脚发出停止位后，如果没有数据等待发送数据缓冲区，发送器置位发送完成标志并进入空闲状态，TxD 逻辑高，等待传输字符。

写 0 到 TE 并不能立即使引脚成为通用 I/O 引脚，必须先完成正在进行的发送过程。这包括正在处理的数据，等待空闲字符和等待间隔符。

13.3.2.1 发送间隔和等待空闲

SCIxC2 的 SBK 用于发送间隔符 （原来用于与电传打字机进行通信初始化）。间隔符是一个字符的逻辑 0 （10 位包含开始位和停止位）。置位 BRK13=1，则间隔符可以达到 13 位的长度。通常，程序等待 TDRE 置位 （表示信息的最后一个字符移到发送移位寄存器），然后向 SBK 位先后写 1 和 0。这样的话，只要移位寄存器可用，就会发送间隔符。如果当间隔符进入移位寄存器 （由波特率时钟同步） SBK 仍为 1，则有另外

一个间隔符在等待。如果接收设备是另一个飞思卡尔半导体的 SCI，则接受的 8 位间隔符全 0，并且产生帧错误 （FE=1）。

当使用空闲线唤醒时，两个信息之间需要一个空闲 （逻辑 1）的字符，以唤醒处于睡眠状态的接收器。通常，程序等待 TDRE 置位 （表示信息的最后一个字符移到发送移位寄存器），然后向 TE 位先后写 0 和 1。这样的话，只要移位寄存器可用，就会发送空闲符。只要移位寄存器的字符没有完成 （TE=0）， SCI 发送器就不会释放对 TxD 引脚的控制。如果移位寄存器有完成的可能性 （TE=0），设置为通用 I/O 控制，则引脚和 TxD 复用，输出逻辑 1。这确保了 TxD 线看起来像正常的空闲线，即使 SCI 在向 TE 写 0 和写 1 之间失去对引脚的控制。

间隔符的长度受 BRK13 和 M 位影响，如表 13-9 所示。

表 13-9. 间隔符长度

BRK13	M	间隔符长度
0	0	10 位
0	1	11 位
1	0	13 位
1	1	14 位

13.3.3 接收功能描述

在该节中，接收器框图 （图 13-3）是全面描述接收器的功能一个指南。然后，更详细地描述数据采样技术 （用于得到接收器的数据）。最后，解释了两种接收器唤醒功能。

通过置位 RXINV=1，接收器输入被反转。置位 SCIxC2 的 RE，接收器使能。字符帧包括一个开始位（逻辑 0）， 8 （或 9）数据位 （首先最低有效位），和停止位 （逻辑 1）。关于 9 位数据模式的信息，参见 13.3.5.1 节 8 位和 9 位数据模式。在以后的讨论中，我们假定 SCI 设置为正常的 8 位数据模式。

接受移位寄存器接收到停止位后，如果接收数据还未满，数据字符传到接收数据寄存器并且接收数据寄存器满 （RDRF）状态标志置位。如果 RDRF 已经置位表明接收数据寄存器 （缓冲区）已经满了，溢出（OR），状态标志置位并且新数据丢失。因为 SCI 接收器时双缓冲，在 RDRF 置位后和读接收数据缓冲区的数据前，程序有一个字符的时间避免接收器溢出。

当程序检测到接收数据寄存器满时 （RDRF=1），读 SCIxD 可以得到接收数据寄存器的数据。 RDRF 通过两步骤的序列（通常在控制接收数据的用户程序中）自动清零，更多关于标志清零的信息参见 13.3.4 节 中断和状态标志。

13.3.3.1 数据采样技术

SCI 接收器使用 16 倍的波特率时钟进行采样。接收器以 16 倍波特率采样逻辑电平，发现 RxD 串行输入引脚上的下降沿。下降沿被定义为 3 个逻辑 1 采样后的逻辑 0 采样。 16 倍波特率时钟把位时间分成 16 段（从 RT1 到 RT16）。当定位了一个下降沿时，有 3 个采样 （RT3、 RT5、 RT7）确保这是开始位而不是噪音。如果 3 个采样中不少于 2 个是 0，则接收器假定与接收字符同步。

然后接收器采样每个位时间，包括开始位和停止位 （在 RT8、 RT9、 RT10 决定改为的电平）。逻辑电平由位时间的采样结果的多数采样决定。采样开始位时，如果 RT3、 RT5、 RT7 这三个中至少两个是 0，则可以认为该位为 0，即使 RT8、 RT9、 RT10 中的一个或全部为 1。如果在字符帧中任何一个位时间 （包括开始位

和停止位）的任何一个采样无法得到一致的逻辑电平，当接收的字符传到接收数据缓冲区时，噪音标志（NF）置位。

下降沿检测逻辑不断检测下降沿，如果检测到了，采样时钟同步到位时间。这提高了接收器在噪音和波特率不匹存在时的可靠性。这不能提高最糟情况的分析，因为有一些字符在字符帧中都不会出现下降沿。

在帧错误的情况下，如果接收的字符不是间隔符，查找下降沿的采样逻辑填充了 3 个逻辑 1，如此以至于几乎可以立即检测一个新的开始位。

在帧错误的情况下，直到帧错误清零，接收器都会被阻止接收新的字符。接受移位寄存器继续工作，但是如果 FE 置位，一个完成的字符不能转移到接收数据缓冲区。

13.3.3.2 接收唤醒

接收器唤醒是一种硬件机制，它允许 SCI 接收器忽略那些供不同 SCI 接收器使用的信息中的字符。在这个系统中，所有的接收器判断每个信息的第一个字符，只要认为信息是提供给不同 SCI 的，则将 SCIxC2 中的接受唤醒 （RWU）置 1。当 RWU 置位，和接收器相关的状态 （除了空闲位 （IDLE），当 RWUID 位置位）禁止置位，因此消除控制不重要字符的软件开支。在信息的最后，或者是下一个信息的开始，所有的接收器自动强制 RWU 为 0，所以所有的接收器按时唤醒，查看下一个信息的第一个字符。

13.3.3.2.1 空闲线唤醒

当 WAKE=0，接收器被设置为空闲线唤醒。在这种模式中，当接收器检测到一个满字符的空闲线电平时， RWU 自动清零。 M 控制位选择 8 位或 9 位数据模式，这决定了多少个空闲位组成了一个满字符时间（加上开始位和停止位共 10 位或 11 位）。

当 RWU 为 1 和 RWUIN 为 0 时，唤醒接收器的空闲条件不置位 IDLE。接收器唤醒并等待下一个信息的第一个字符 （将会置位 RDRF，如果允许可以产生一个中断），当 RWUID 为 1 时，无论 RWU 是 0 还是 1，任何空闲条件置位 IDLE 标志并产生一个中断 （如果允许）。

空闲线类型 （ILT）控制位选择两种检测空闲线方法中的一种。 ILT=0，空闲位从开始位开始计数，停止位和字符后面的逻辑 1 的个数组成了空闲线满字符。 ILT=1，空闲位不是从开始位而是停止位，所以空闲线检测不受前一个信息的最后的字符的数据的影响。

13.3.3.2.2 地址标志唤醒

当 WAKE=1，接收器被设置为地址标志唤醒。在这种模式中，当接收器检测到接收数据的大多数有意义位是逻辑 1 时 （M=0，第 8 位； M=1，第 9 位）， RWU 自动清零。

地址标识唤醒允许信息包含空闲位，但是需要 MSB 保留为地址帧使用。在接收到停止位并且置位 RDRF 标志前，地址帧的 MSB 逻辑 1 清零 RWU 位。在这种情况下，即使接收器在大多数字符时间都在休眠， MSB 字符仍被接收。

13.3.4 中断和状态标志

SCI 系统有三个独立的中断向量，减少了分析中断原因的软件编程。一个中断向量和发送器的 TDRE 和 TC 事件相关。另一个中断向量和接收器的 RDRF、 IDLE、 RXEDGIF 和 LBKDIF 事件相关。第三个向量用于 OR、 NF、 FE 和 PD 错误条件。通过本地中断使能掩码，这十个中断源可以独立掩码。当本地掩码被清零，禁止中断请求的产生时，这个标志可以通过软件轮询的方式得到。

SCI 发送器有两个状态标志，可以产生硬件中断请求。发送数据寄存器空 （TDRE）表示什么时候发送数据缓冲区有空间可以写另一个发送字符到 SCIxD。如果发送中断使能 （TIE）位置位，只要 TDRE=1，就会产生一个硬件中断请求。发送完成 （TC）表示所有的数据，前导符和间隔符都发送完成，发送器空闲 （TxD 处于无效电平）。该标志位通常用于有调制解调器的系统，用于决定何时关闭调制解调器是安全的。如果发送完成中断使能位 （TCIE）置位，只要 TC=1，就会产生一个硬件中断请求。如果 TIE 或 TCIE 本地中断掩码是 0，软件轮询可以代替硬件中断请求，可以用于检测 TDRE 和 TC 状态位。

当程序检测到接收数据寄存器满 （RDRF=1），程序读 SCIxD 可以得到接收数据寄存器的数据。当 RDRF=1 并且读了 SCIxD 后，读 SCIxS1 可以清零 RDRF 标志。

当使用轮询方式时，这个顺序可以很自然的满足用户程序的一般过程。如果使用硬件中断方式， SCIxS1 必须在中断服务例程 （ISR）中读。通常这必须在 ISR 中完成已检测接收错误，所以这个顺序可以自动满足。

当 RxD 线在一段时间保持空闲， IDLE 状态标志包含了可以避免重复置位的情况。当 IDLE=1 并且读了 SCIxD 后，读 SCIxS1 可以清零 IDLE。 IDLE 清零后，直到接收器至少接收了一个新字符并且 RDRF 置位，才能再次置位。

如果在导致 RDRF 置位的接收字符中检测到了相应的错误，错误标志——噪音标志 （NF），帧错误 （FE）和奇偶校验错误标志 （PF）将会和 RDRF 同时置位。在溢出情况，这些标志不置位。

如果 RDRF 置位，当一个新字符准备从接收移位寄存器到接收数据缓冲区，溢出标志 （OR）置位，和数据相关的 NF、 FE 或 PF 条件丢失。

在任何时候， RxD 串行数据输入引脚上的有效边沿会导致 RXEDGIF 标志置位。向 RXEDGIF 标志写 1，可以清零 RXEDGIF。这个功能不依赖于接收器被允许 （RE=1）。

13.3.5 其他 SCI 功能

下面描述其他的 SCI 功能。

13.3.5.1 8 位和 9 位数据模式

通过置位 SCIxC1 的 M 控制位， SCI 系统 （发送器和接收器）可以被设置为 9 位数据模式。在 9 位数据模式中，第九位数据位在 SCI 数据寄存器的最高有效位的左边。对于发送数据缓冲区，该位存储在 SCIxC3 的 T8。对于接收器，该位存储在 SCIxC3 的 T8。

为了一致地写发送数据缓冲区，在写 SCIxD 以前写 T8 位。

如果要发送的新字符的第九位的值和前一个字符相同，没有必要再写 T8。当数据从发送数据缓冲区传到发送移位寄存器中时，在数据从 SCIxD 传到移位寄存器的同时， T8 的值被复制。

9 位数据模式通常用于连接奇偶校验位，可以允许 8 位的数据加上第九位的奇偶校验位。或用于地址标志唤醒，所以第九位也可以作为唤醒位。在定制协议中，第九位也作为一个由软件控制的掩码器。

13.3.5.2 停止模式操作

在所有的停止模式中， SCI 模块的时钟暂停。

在 stop1 和 stop2 模式中，所有的 SCI 寄存器数据丢失。当从这两种停止模式中恢复时，所有的寄存器必须重新初始化。在 stop3 模式中， SCI 模块寄存器不会受影响。

在 stop3 模式中，接收输入有效边沿检测电路仍然有效，但是在 stop2 模式下无效。如果中断没有被屏蔽（RXEDGIE=1），接收输入上的一个可以将 CPU 从 stop3 模式中唤醒。

注意，因为时钟暂停，当从停止模式 （仅在 stop3 模式中）中退出， SCI 模块将重新开始操作。当有字符正在被发送出或接收进 SCI 模块，软件应该确保不进入停止模式。

13.3.5.3 循环模式

当 LOOPS=1 时，同一个寄存器的 RSRC 位选择循环模式 （RSRC=0）或单线模式 （RSRC=1）。循环模式有时用于检测软件，独立于外部系统的连接，可以帮助分析系统问题。在这个模式中，发送器的输出在内部连接到接收器的输入， RxD 不使用，所以它作为通用 I/O 引脚。

13.3.5.4 单线操作

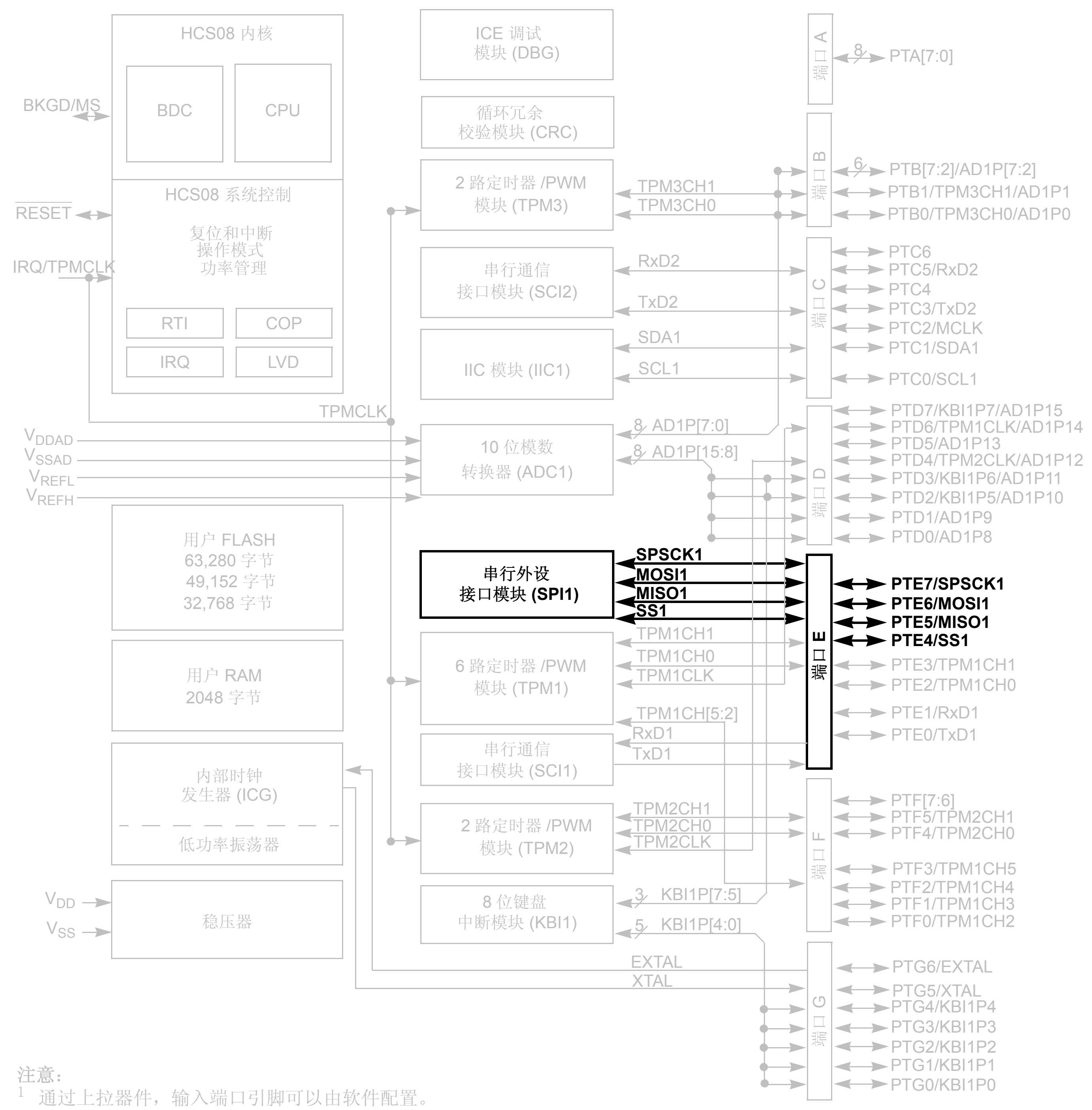
当 LOOPS=1 时，同一个寄存器的 RSRC 位选择循环模式 （RSRC=0）或单线模式 （RSRC=1）。单线模式半双工的串行连接。接收器在内部连接到发送器的输出 TxD 引脚。 RxD 引脚不使用，作为通用 I/O 引脚。

在单线模式中， SCIxC3 的 TXDIR 位控制 TxD 引脚上数据的方向。当 TXDIR=0， TxD 引脚作为接收器的输入并且发送器和 TxD 引脚临时断开，所以一个外部设备可以向接收器发送数据。当 TXDIR=1， TxD 作为接收器的输出。在单线模式中，从发送器到接收器的内部环路使接收器接收发送器所发送的字符。

第 14 章
串行外设接口 (S08SPIV3)

14.1 引言

MC9S08AC60 系列提供一个串行外设接口（SPI）模块。如需了解 SPI 电参数信息，请参见附录 A 电气特性和时序规范。



注意：

1 通过上拉器件，输入端口引脚可以由软件配置。

2 如果 IRQ 使能 (IRQPE = 1)，引脚包含软件可配置的上拉 / 下拉器件。如果选择上升边沿检测，则下拉使能 (IRQEDG = 1)。

3 引脚包含集成上拉器件。

4 PTD3、PTD2、PTD7 和 PTG4 既包含上拉也包含下拉器件。当 KBI 使能时 (KBIPEn = 1)，且选择了上升边沿 (KBEDGn = 1)，则下拉使能 。

5 TPMCLK、 TPM1CLK 和 TPM2CLK 选项可通过软件配置； 复位后， TPM1CLK、 TPM2CLK 和 TPMCLK 分别提供给 TPM1、 TPM2 和 TPM3。

图 14-1. 显亮 SPI 模块及其引脚的 MC9S08AC60 系列模块图

14.1.1 特性

SPI 模块特性如下：

- 支持主或者从工作方式
- 支持全双工或半双工
- 发送比特率可编程
- 支持双缓存发送和接收
- 支持串行时钟相位和极性可选
- 支持从设备选择输出
- 支持 MSB 优先移位或 LSB 优先移位

14.1.2 模块结构图

本节介绍 SPI 模块的结构图，图中显示了 SPI 系统连接、 SPI 模块的内部组成以及控制主模式比特率的 SPI 时钟分频器。

14.1.2.1 SPI 系统结构图

在图 14-2 中，两个 MCU 的 SPI 模块以主从方式连接。所有 SPI 数据传送由主设备发起。在一个发送过程中，主机送出数据 （从 MOSI 脚）给从机，同时接收来自从机的数据 （从 MISO 脚）。两个 SPI 系统中 SPI 移位寄存器的数据通过这种方式有效地进行交换。 SPSCK 信号是主机的一个时钟输出，同时作为从机的一个时钟输入。从机设备必须由从机选择输出脚上的低电平来被选中。在这种系统中，主机设备设置它的 $\overline{\text{SS}}$ 引脚作为一个可选择的从机选择输出。

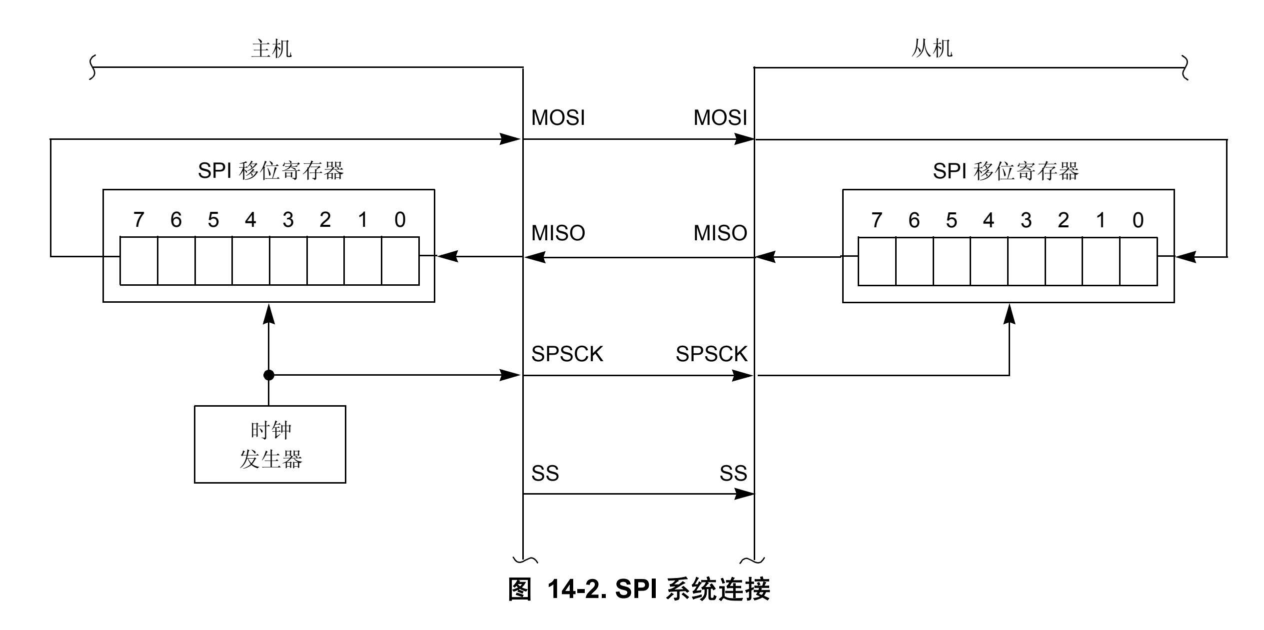


图 14-2. SPI 系统连接

SPI 系统通常用于连接简单的移位寄存器用于增加输入或输出端口，或者连接一个小型的外围设备，如串行 A/D， D/A 转换器。虽然图 14-2 中的系统在两个 MCU 之间进行双向数据交换，但在实际应用中，更普遍的是主从机之间的单向数据交换。

14.1.2.2 SPI 模块框图

图 14-3 是 SPI 模块结构图。 SPI 移位寄存器是 SPI 模块的核心。数据写入双缓存发送器（写给 SPID）并在发送开始时传送给 SPI 移位寄存器。在移入一个字节数据后，数据被送入双缓存接收器（从 SPID 读出）来读出。引脚复用逻辑控制 MCU 引脚和 SPI 模块之间的连接。

当把 SPI 设置为主机，该时钟输出被发送到 SPSCK 引脚，移位寄存器输出发送到主出从入引脚（MOSI），移位寄存器的输入来自主入从出引脚（MISO）。

当把 SPI 作为从机， SPSCK 引脚作为 SPI 的时钟输入，移位寄存器输出发送到主入从出引脚（MISO），移位寄存器输入来自主出从入引脚（MOSI）引脚。

在外部的 SPI 系统中，所有 SPSCK 引脚简单地相互连接，所有主入从出引脚在一起，主出从入引脚在一起。外围设备对这些引脚的命名通常略有不同。

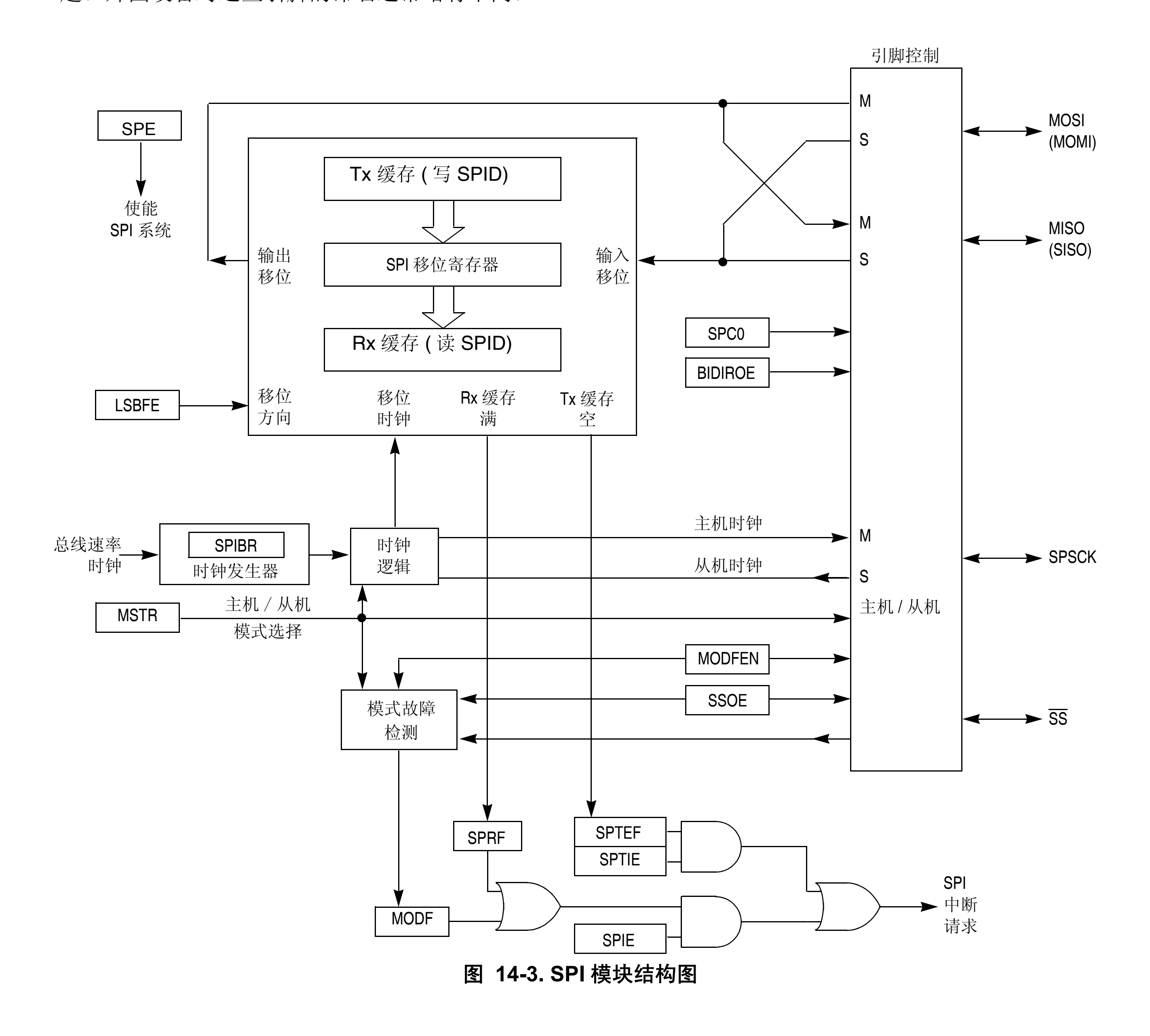


图 14-3. SPI 模块结构图

14.1.3　SPI 波特率发生器

如图 14-4 所示， SPI 波特率发生器采用总线时钟作为时钟源。三个分频位 （SPPR2 :SPPR1:SPPR0）在 1、2、3、4、5、6、7 和 8 中选择一个作为分频约数。该三位速率位（SPPR2 :SPPR21:SPPR0）组合在 2、 4、 8、 16、 32、 64 和 128， 256 中选择分频系数获得 SPI 内部主机模式位率时钟。

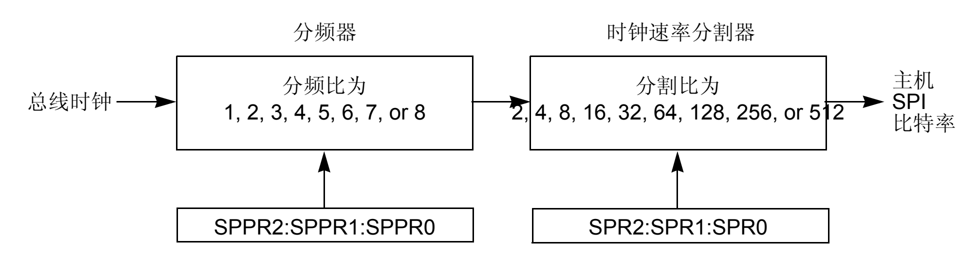


图 14-4. SPI 波特率发生器

14.2　外部信号描述

SPI 可复用四个端口引脚。这些引脚的功能取决于 SPI 控制位的设置。当 SPI 禁止时 （SPE = 0），这四个引脚作为通用 I/O 端口引脚，不受 SPI 控制。

14.2.1　SPSCK — SPI 串行时钟

当 SPI 工作于从机方式，该引脚为串行时钟输入。当 SPI 工作于主机方式，该引脚为串行时钟输出。

14.2.2　MOSI — 主出从入引脚

当 SPI 工作于主机方式并且 SPI 引脚控制的零脚 （SPC0）为 0 （非双向模式），该引脚为串行数据输出。

当 SPI 工作于从机方式且 SPC0 = 0 时，该引脚为串行数据输入。如果 SPC0 = 1 选择单线双向模式，并选择为主机模式，该引脚转变为双向数据 I/O （MOMI）引脚。同样，双向模式输出使能位决定该引脚是作为输出 （BIDIROE = 1）还是输入 （BIDIROE = 0）。如果 SPC0 = 1 并选择从机模式，该引脚作为通用 I/O 端口引脚，不被 SPI 使用。

14.2.3　MISO — 主入从出引脚

当 SPI 工作于主机方式且 SPI 引脚控制零脚 SPC0 为 0 （非双向模式），该引脚为串行数据输入。当 SPI 工作于从机方式且 SPC0 = 0 时，该引脚为串行数据输出。如果 SPC0 = 1 选择单线双向模式，同时选择从机模式，该引脚为双向数据 I/O 引脚 （SISO）。同时，双向模式使能位决定该引脚为输出 （BIDIROE = 1）还是输入 （BIDIROE = 0）。如果 SPC0 = 1 且选择主机模式，该引脚作为一个通用 I/O 端口引脚，不被 SPI 使用。

14.2.4　$\overline{\text{SS}}$ — 从机选择引脚

当 SPI 工作于从机方式时，该引脚为低电平从机选择输入脚。当 SPI 工作于主机方式且模式故障功能禁止 （MODFEN = 0），该引脚作为一个通用 I/O 端口引脚，不被 SPI 使用。当 SPI 工作于主机方式且 NODFEN = 1 时，从机选择输出使能位决定该引脚是作为模式错误输入 （SSOE = 0）还是作为从机选择输出（SSOE = 1）。

14.3 操作模式

14.3.1 停止模式中的 SPI

SPI 在所有停止模式中都被禁止，而无论在执行停止指令之前作了什么设置。在停止模式 1 和停止模式 2 期间， SPI 模块被完全关断。当从停止模式 1 和停止模式 2 中唤醒时， SPI 模块将进行复位。在停止模式 3 期间， SPI 模块时钟被禁止，而所有寄存器的内容被保持。如果通过复位退出停止模式 3， SPI 将进入复位状态。如果通过中断退出停止模式 3， SPI 将维持进入停止模式 3 时的状态。

14.4 寄存器定义

SPI 有五个 8 位寄存器来选择 SPI，控制波特率，报告 SPI 状态，和发送 / 接受数据。

关于如何分配 SPI 寄存器的绝对地址，请参阅存储器章节的直接页寄存器表。本节通过名称来检索寄存器和控制位，飞斯卡尔定义的头文件翻译这些名称为相应的绝对地址。

14.4.1 SPI 控制寄存器 1 （SPIC1）

该读 / 写寄存器包括 SPI 使能控制，中断使能和配置选项。

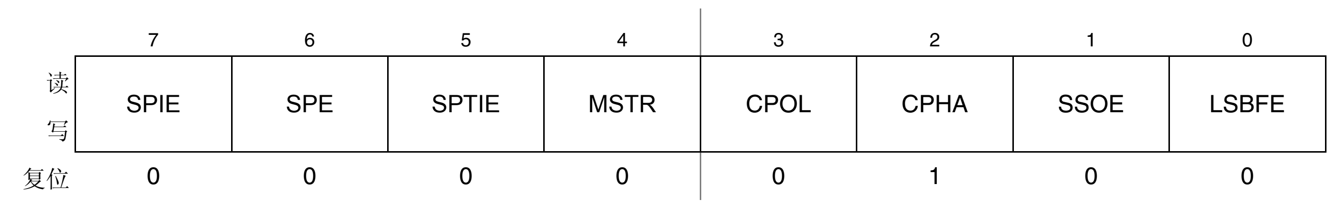


图 14-5. SPI 控制寄存器 1 （SPIC1）

表 14-1. SPIC1 位描述

位	描述
7 SPIE	SPI 中断使能位 （SPRF 和 MODF）— 这是一个中断使能位，它用来控制 SPI 接收缓存满 （SPRF）和模式错误 （MODF）事件。 0 禁止 SPRF 和 MODF 中断 （使用轮询检测） 1 当 SPRF 或 MODF 为 1 时，请求硬件中断
6 SPE	SPI 系统允许位 — 禁止 SPI，并停止进程中任何传送，清除数据缓存，初始化内部状态机。 SPRF 被清零，并且 SPTEF 被置位来标识 SPI 传送数据缓存为空。 0 SPI 系统禁止 1 SPI 系统允许
5 SPTIE	SPI 发送中断允许位 — 该位使能用于 SPI 发送缓存空中断 0 禁止 SPTE 中断 （使用轮询检测） 1 当 SPTE 为 1 时，请求硬件中断
4 MSTR	主机 / 从机模式选择位 0 SPI 模块配置为从机 SPI 设备 1 SPI 模块配置为主机 SPI 设备
3 CPOL	时钟极性位 — 该位有效放置一个变极器串联来自主机 SPI 或给从机 SPI 的时钟信号。详细信息参阅 14.5.1 节 SPI 时钟格式 0 SPSCK 空闲时为低电平 1 SPSCK 空闲时为高电平

表 14-1. SPIC1 位描述

位	描述
2 CPHA	时钟相位位 — 该位为不同类型的同步串行外设选择一种时钟格式。详情参阅 14.5.1 节 SPI 时钟格式 0 SPSCK 第一次沿跳变发生在 8 个数据发送周期的第一个周期的中期 1 SPSCK 第一次沿跳变发生在 8 个数据发送周期的第一个周期的开始
1 SSOE	从机选择输出使能位 — 该位用于连接 SPCR2 模式故障使能位 （MODFEN）和主机 / 从机控制位 （MSTR）决定表 14-2 中 SS 脚功能
0 LSBFE	LSB 优先 （移位方向） 0 开始时即以最多显著位进行 SPI 串行数据传送 1 开始时即以最少显著位进行 SPI 串行数据传送

表 14-2. $\overline{SS}$ 引脚功能

MODFEN	SSOE	主机模式	从机模式
0	0	通用 I/O （非 SPI）	从机选择输入
0	1	通用 I/O （非 SPI）	从机选择输入
1	0	$\overline{SS}$ 引脚输入用于模式故障	从机选择输入
1	1	$\overline{SS}$ 引脚自动输出	从机选择输入

注意

当位改变为 CPHA 位时，确保 SPI 不能同时被禁止 （SPE=0）。这些改变必须单独操作，否则会发生异常。

14.4.2 SPI 控制寄存器 2 （SPIC2）

该读 / 写寄存器用于控制 SPI 系统选项特征。位 7， 6， 5， 2 是无效的，一直读为 0

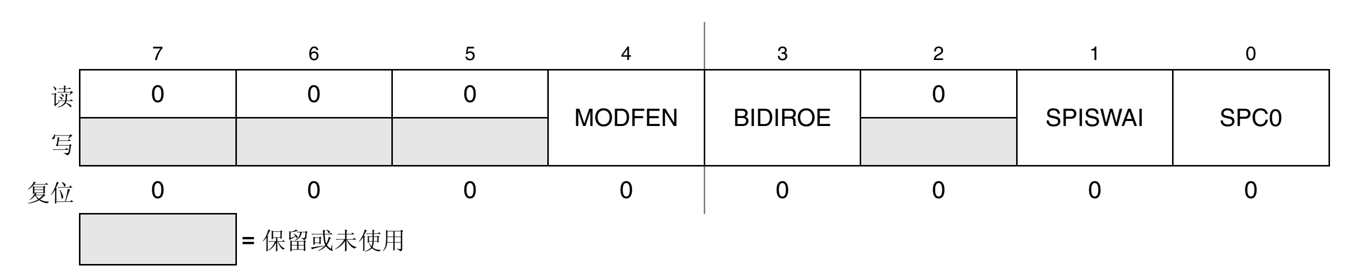


图 14-6. SPI 控制寄存器 2 （SPIC2）

表 14-3. SPIC2 寄存器位描述

位	描述
4 MODFEN	主机模式 - 故障功能使能位 - 当 SPI 工作于从机模式，该位无效，（SS 引脚是从机选择输入引脚）在主机模式下该位决定 SS 引脚如何使用 （详情参阅表 14-2） 0 模式故障功能禁止，主机 SS 引脚作为通用 I/O 引脚不受 SPI 控制 1 模式故障功能允许，主机 SS 引脚作为模式故障输入或从机选择输出
3 BIDIROE	双向模式输出允许—当设置 SPI 引脚控制 (SPC0)=1，允许双向模式， BIDIROE 将决定该 SPI 数据输出是否采用单线双向 SPI I/O 引脚。根据 SPI 工作于主机模式还是从机模式，它使用 MOSI(MOMI) 或 MISO(SISO) 引脚中的一个，作为 SPI 数据 I/O 引脚。当 SPC0=0 时， BIDIROE 无效。 0 输出驱动禁止， SPI 数据 I/O 引脚作为输入引脚 1 SPI I/O 引脚允许作为输出引脚
1 SPISWAI	在等待模式下 SPI 停止 0 SPI 时钟继续工作于等待模式 1 当 MCU 进入等待模式时， SPI 时钟停止
0 SPC0	SPI 引脚控制 0—SPC0 设置单线双向模式。如果 MSTR=0 （从机模式），该 SPI 使用 MISO （SISO）引脚用于双向 SPI 数据传输。如果 MSTR=1 （主机模式），该 SPI 使用 MOSI （MOMI）引脚用于双向 SPI 数据传输。当 SPC0=1， BIDIROE 用来允许或禁止用于单线双向 SPI I/O 引脚的输出驱动。 0 SPI 使用独立引脚用于数据输入，输出 1 SPI 配置为单线双向模式

14.4.3 SPI 比特率寄存器 （SPIBR）

该寄存器为工作于主机方式的 SPI 设置分频器和位率因子，可在任何时候被读或写。

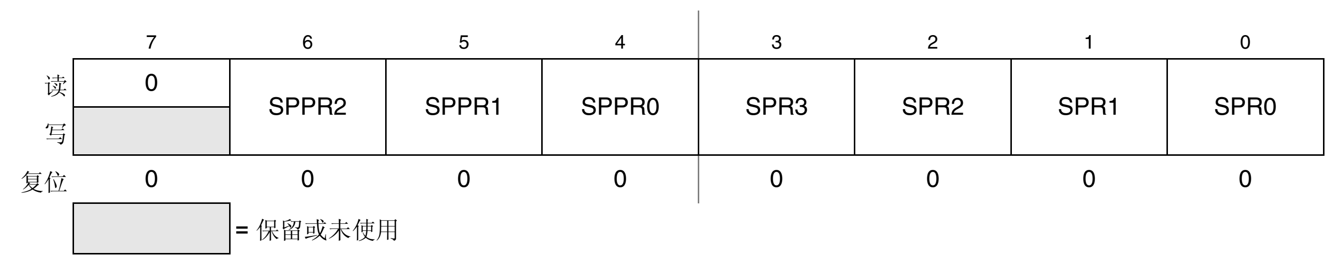


图 14-7. SPI 比特率寄存器 （SPIBR）

表 14-4. SPIBR 寄存器位描述

位	描述
6:4 SPPR[2:0]	SPI 波特率分频因子—这三位为 SPI 波特率分频器选择八个因子中的一个，如表 14-5 所示。该分频器的输入是总线速率时钟 （BUSCLK）。该分频器的输出驱动 SPI 波特率分割器的输入 （见图 14-4）
2:0 SPR[2:0]	SPI 波特率因子—这三位为 SPI 波特率分割器选择表 14-6 中的八个因子中的一个。该分割器的输入来自 SPI 波特率分频器 （见图 14-4）。该分割器的输出为主机模式下的 SPI 位率时钟。

表 14-5. SPI 波特率分频器因子

SPPR2:SPPR1:SPPR0	分频器因子
0:0:0	1
0:0:1	2
0:1:0	3

表 14-5. SPI 波特率分频器因子 (续)

SPPR2:SPPR1:SPPR0	分频器因子
0:1:1	4
1:0:0	5
1:0:1	6
1:1:0	7
1:1:1	8

表 14-6. SPI 波特率因子

SPPR2:SPPR1:SPPR0	位率因子
0:0:0	2
0:0:1	4
0:1:0	8
0:1:1	16
1:0:0	32
1:0:1	64
1:1:0	128
1:1:1	256

14.4.4 SPI 状态寄存器 （SPIS）

该寄存器有三个只读状态位。位 6， 3， 2， 1， 0 未使用，通常读为 0。对它们写无效且无影响。

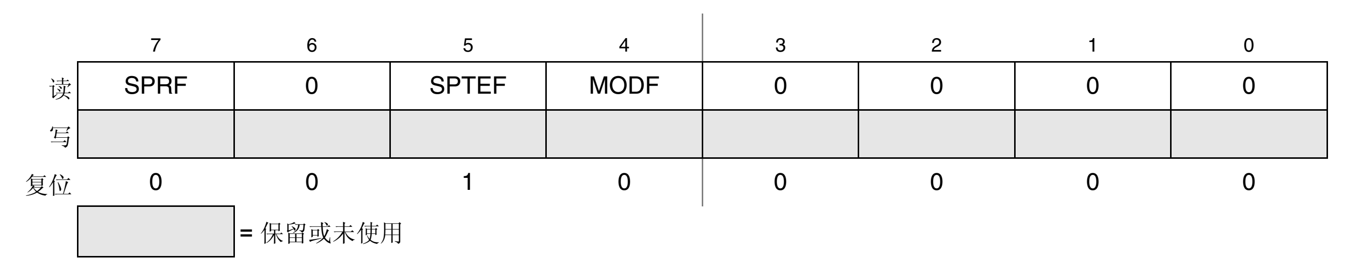
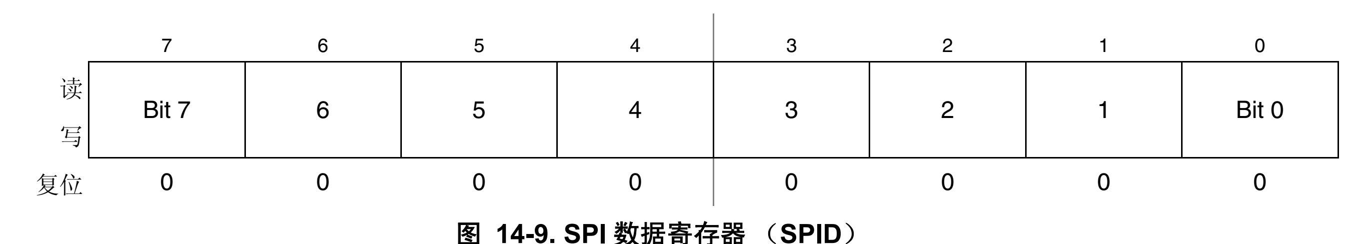


图 14-8. SPI 状态寄存器 （SPIS）

表 14-7. SPI 状态寄存器 （SPIS）

位	描述
7 SPRF	SPI 读缓冲区满标识—SPRF 在 SPI 完成数据传送时被置位，表明收到的数据可能来自 SPI 数据寄存器（SPID）。 SPRF 通过在设置时读自己可以清零，然后读 SPI 数据寄存器。 0 在接受数据缓存中无可用数据 1 在接受数据缓存中有数据
5 SPTEF	SPI 发送缓冲区空标识—当发送数据缓冲区有空间时该标识位被置位。它可通过读 SPIS 来清零，然后写一个数据值给 SPID 中的发送缓冲区。写数据给 SPID 之前必须读到 SPIS 的 SPTEF=1，否则写 SPID 将被忽略。如果 SPIC1 中的 SPTIE 位也被置位，SPTEF 产生一个 SPTEF CPU 中断请求。当一个字节的数据从发送缓冲区里传送到发送移位寄存器时， SPTEF 被自动置位。对于闲置的 SPI （无数据在发送缓冲区或移位寄存器中并且无正在进行的发送），被写到 SPID 的数据被立即发送到移位寄存器，所以 SPTEF 在两个总线周期内被置位允许第二个 8 位数据值进入发送缓冲区排队。完成移位寄存器中的值的传送后，来自发送缓冲区的排队值将自动移入到移位寄存器， SPTEF 将被置位以表明发送缓冲区里还有空间存放新数据。如果发送缓冲区里没有新数据， SPTEF 保持置位，无数据从缓冲区移到移位寄存器。 0 SPI 发送缓冲区非空 1 SPI 发送缓冲区空
4 MODF	主机模式故障标识—如果 SPI 设置为主机方式且从机选择输入脚为低， MODF 被置位，表明其它一些 SPI 设备也被设置为主机方式。仅当 MSTR=1， MODFEN=1，并且 SSOE=0 时， SS 引脚用于模式故障输入；否则 MODF 不会被置位。当 MODF 为 1 时，通过读它可以清零，然后写 SPI 控制寄存器 1 （SPIC1） 0 无模式故障错误 1 模式故障错误检测

14.4.5 SPI 数据寄存器 （SPID）



	7	6	5	4	3	2	1	0
读 写	Bit 7	6	5	4	3	2	1	Bit 0
复位	0	0	0	0	0	0	0	0

图 14-9. SPI 数据寄存器 （SPID）

读该寄存器返回数据缓冲区里的数据。写该寄存器将写数据到发送数据缓冲区。当 SPI 配置为主机方式时，写数据到发送数据缓冲区表明一个 SPI 发送开始。

不要将数据写入发送数据缓冲区除非该 SPI 发送缓冲区空标识 （SPTEF）置位，表明发送缓冲区里有空间来排队一个新的发送字节。

在 SPRF 被置位后到在另一次传送结束之前，可以从 SPID 读取数据。在一个新的传送结束之前，若读取接收数据缓冲区外的数据失败，则接收溢出并且新传送来的数据将丢失。

14.5 功能描述

通过检测 SPI 发送缓冲区空标识 （SPTEF=1）开始一个 SPI 传送，然后写一个字节数据给主机 SPI 数据寄存器 （SPID）。当 SPI 移位寄存器可用时，该字节数据从发送数据缓冲区移到移位寄存器， SPTEF 被置位标识缓冲区里还有空间可供其他发送字符排队， SPI 串行发送开始。

在 SPI 发送数据期间，数据在 MISO 脚上的一个 SPSCK 时钟边沿被采样 （读取）和转移，在稍后的一个半 SPSCK 时钟周期，改变 MOSI 引脚上的位值。在八个 SPSCK 时钟周期后，主机移位寄存器中的数据通过 MOSI 引脚发送给从机，同时八位的数据通过 MISO 引脚从从机转移到主机移位寄存器。在这次传送过程

最后，接收到的数据从移位寄存器移到接收数据缓冲区里；同时 SPRF 被置位以表明可通过读 SPID 来获取数据。如果另一个字节的数据在发送缓冲区里等待，它将被转移到移位寄存器， SPTEF 被置位且一个新的传送开始。

通常 SPI 数据首先被发送最高有效位 （MSB）。如果最低有效位优先 （LSBFE）被置位，则优先发送最小位。

当 SPI 被配置为从机方式，其 $\overline{SS}$ 引脚在发送开始前必须被拉低并在整个传送过程中保持低电平。如果 CPHA=0，在连续传送时 $\overline{SS}$ 引脚被置位逻辑 1. 如果 CPHA=1，在连续传送时 $\overline{SS}$ 引脚保持低电平。详情参阅 14.5.1 节 SPI 时钟格式。

因为发送器和接收器都是双缓冲，所以除了当前正在被移出的字节外的第二个字节可以排队进入发送数据缓冲区；并且当一个新数据正在移入的同时，先前接收到的数据可以仍然处于接收数据缓冲区。当发送缓存区有空间用于新的字符时，通过 SPTEF 标识。当接收数据缓冲区有接收到的数据时，通过 SPRF 来标识。接收到的数据必须在下个传送结束之前从接收缓冲区里读出，否则产生一个接受溢出错误。

在接收溢出情况下，新数据将丢失因为接收缓冲区仍然保存先前的数据而没有准备接收新数据。目前，这种溢出情况没有标识，因此应用系统的设计者必须确保以前的数据在一个新的传送过程开始之前已经被读出接收缓冲区。

14.5.1 SPI 时钟格式

为支持来自不同厂商不同种类的同步串行外设， SPI 系统含有一个时钟极性位 （CPOL）和一个时钟相位控制位 （CPHA）来从四种数据传输时钟形式中选择一种。 CPOL 选择性的插入一个变极器和时钟串联。 CPHA 在时钟和数据之间选择两个不同时钟相位关系。

图 14-10 为 CPHA=1 时的时钟形式。在该图的顶部的八位用来参考，位 1 在第一个 SPSCK 的边沿开始，位 8 在第 16 个 SPSCK 边沿后的一个半 SPSCK 周期结束。 MSB 优先和 LSB 优先排列显示的 SPI 数据位的顺序由 LSBFE 中的设置确定。 SPSCK 极性的两种不同变化被显示出来，但根据 CPOL 的值，只有一种波形应用于具体的传送。 SAMPLE IN 波形应用于一个从机的 MOSI 引脚输入或一个主机的 MISO 引脚输入。 MOSI 波形应用于来自主机的 MOSI 引脚输出，而 MISO 波形应用于来自从机的 MISO 输出。该 $\overline{SS}$ 输出波形应用于来自主机的从机选择输出 （假设 MODFEN=SSOE=1）。该主机的 $\overline{SS}$ 引脚输出在传送开始前的半个 SPSCK 周期开始为低电平，在第八个位结束后变为高电平。 $\overline{SS}$ 输入波形应用于一个从机的从机选择输入。

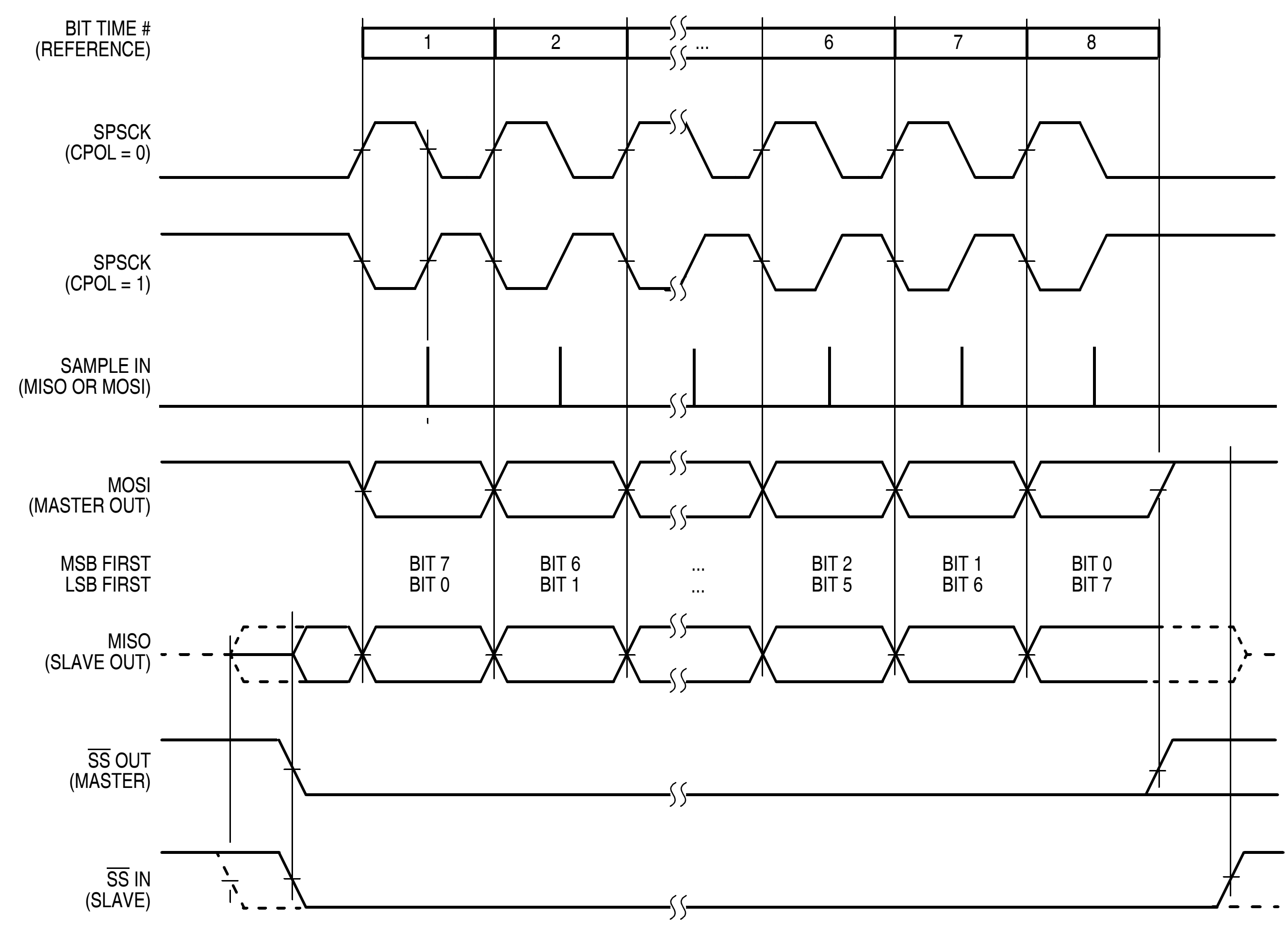


图 14-10. SPI 时钟格式 （CPHA=1）

当 CPHA=1 时，$\overline{SS}$ 引脚走低时从机开始驱动其 MISO 引脚输出，当直到第一个 SPSCK 边沿数据才被识别。在第一个 SPSCK 边沿从移位寄存器移出数据的第一位到主机的 MOSI 输出引脚和从机的 MISO 输出引脚。下个 SPSCK 边沿主机和从机分别采样 MISO 和 MOSI 输入引脚上的数据。在第三个 SPSCK 边沿，该 SPI 移位寄存器移出一位，移入该位的值恰是移位寄存器采样和移位得到的第二个数据位值分别输给主机和从机的 MOSI 和 MISO 输出引脚。当 CPHA=1 时，该从机的 $\overline{SS}$ 引脚输入在传送期间不需要变为高电平。

图 14-11 显示 CPHA=0 时的时钟格式。八个位时间显示在图的顶部以供参考。当从机被选中 （$\overline{SS}$ 引脚输入拉底）时位 1 开始，在最后的 SPSCK 边沿位 8 结束。 MSB 优先和 LSB 优先显示根据 LSBFE 中的设置确定的 SPI 数据位的顺序。 SPSCK 极性的两种变化都被显示，但是只有其中的一种波形应用于具体的传送过程，该波形取决于 CPOL 的值。 SAMPLE IN 波形应用于一个从机的 MOSI 引脚输入或一个主机的 MISO 输入。 MOSI 波形应用于来自主机的 MOSI 引脚输出， MISO 波形应用于来自从机的 MISO 引脚输出。 $\overline{SS}$ OUT 波形应用于来自主机 （假设 MODFEN=SSOE=1）的从机选择输出。该主机的 $\overline{SS}$ 引脚输出在传送的第一个位时间开始时拉低，在八个传送位时间之后的一个半 SPSCK 周期被拉高。 $\overline{SS}$IN 引脚波形应用于一个从机的从机选择输入。

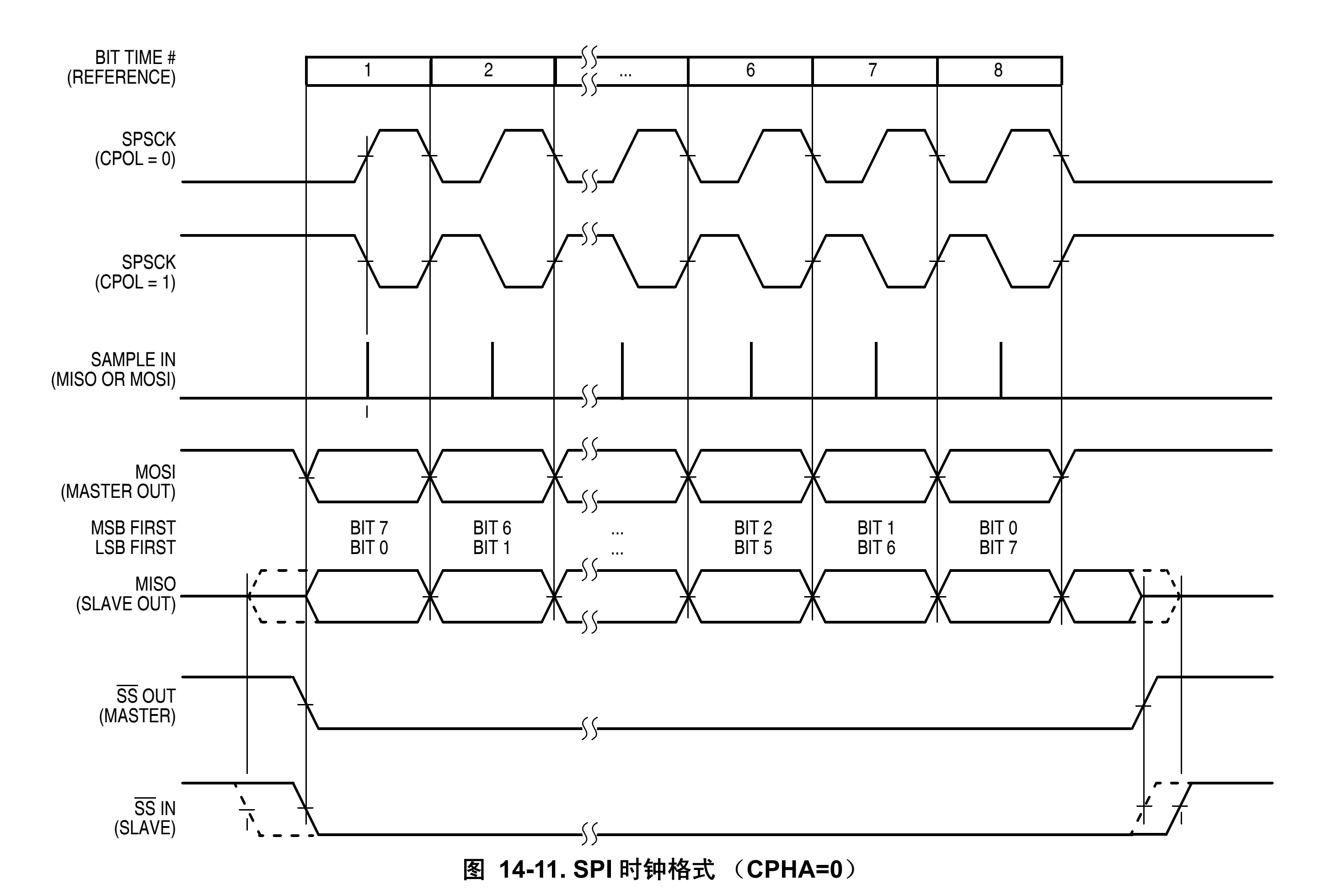


图 14-11. SPI 时钟格式（CPHA=0）

当 CPHA=0，$\overline{SS}$ 引脚为低时，从机开始用第一个数据位值来驱动它的 MISO 引脚输出（由 LSBFE 确定为 MSB 或 LSB）。第一个 SPSCK 边沿时主机和从机各自采样 MISO 和 MOSI 引脚上的数据位值。在第二个 SPSCK 边沿，该 SPI 移位寄存器移出一位，移入该位的值恰是移位寄存器采样和移位得到的第二个数据位值分别输给主机和从机的 MOSI 和 MISO 输出引脚。当 CPHA=0 时，该从机的 $\overline{SS}$ 引脚在传送时必须为高电平。

14.5.2 SPI 中断

与 SPI 系统相关的标识位有三个，包括两个中断屏蔽位和一个中断向量。SPI 中断屏蔽使能位（SPIE）允许来自 SPI 接收缓存满标识（SPRF）中断和模式故障标识（MODF）中断。SPI 发送中断使能屏蔽位（SPTIE）允许来自 SPI 发送缓存空标识（SPTEF）中断。当这些标识位中的任一个被置位并且相应的中断屏蔽位被置位时，一个硬件中断将被发送给 CPU。如果中断屏蔽位被清零，软件可以轮询相关的标识位来代替使用中断。该 SPI 中断服务例程（ISR）检查其标识位引发的中断的事件。在中断服务例程返回之前应该清零这些标识位。

14.5.3 模式故障检测

当一个主机 SPI 设备检测到 $\overline{SS}$ 引脚上的错误时 （假设该 $\overline{SS}$ 引脚被设置成模式故障输入信号）时，模式故障发生，模式故障标识位被置位。当 MSTR=1 时该 $\overline{SS}$ 引脚被配置为模式故障输入信号，模式故障使能位被置位 （MODFEN=1），从机选择输出使能位被清零 （SSOE=0）。

模式故障检测特性可被用于多 SPI 设备主机的系统。当主机的 $\overline{SS}$ 引脚为低时，错误被检测到，此时其它 SPI 设备正试图将这个主机作为从机来访问。这可能导致输出驱动冲突，因此当这样的一个错误被检测到时，模式故障逻辑可以停止所有的 SPI 输出驱动器。

当一个模式故障被检测到时， MODF 被置位且 MSTR 被清零以改变 SPI 配置回从机模式。 SPSCK、 MOSI、 MISO （如果不是双向模式）引脚上的输出驱动被禁止。

通过读 MODF 可清零 MODF，然后，写 SPI 控制寄存器 （SPIC）。在把 SPI 改回到主机模式之前用户软件应确保错误已被纠正。

第 15 章
定时器 /PWM (S08TPMV3)

15.1 引言

MC9S08AC60 系列包括 3 个独立的定时器 /PWM （TPM）模块。这些模块在每个通道上支持传统的输入捕捉、输出比较或缓冲边沿对齐脉宽调制 （PWM) 功能。MC9S08AC60 系列的定时器系统包括 6 路 TPM1、 2 路的 TPM2 和 2 路的 TPM3 各一个。

每一个 TPM 中的控制位将该定时器的所有通道配置为中心对齐的 PWM。在各个 TPM 中，定时功能基于单独的 16 位计数器，用预分频器和模数功能控制时间参考源的频率及范围 （溢出之间的时长）。

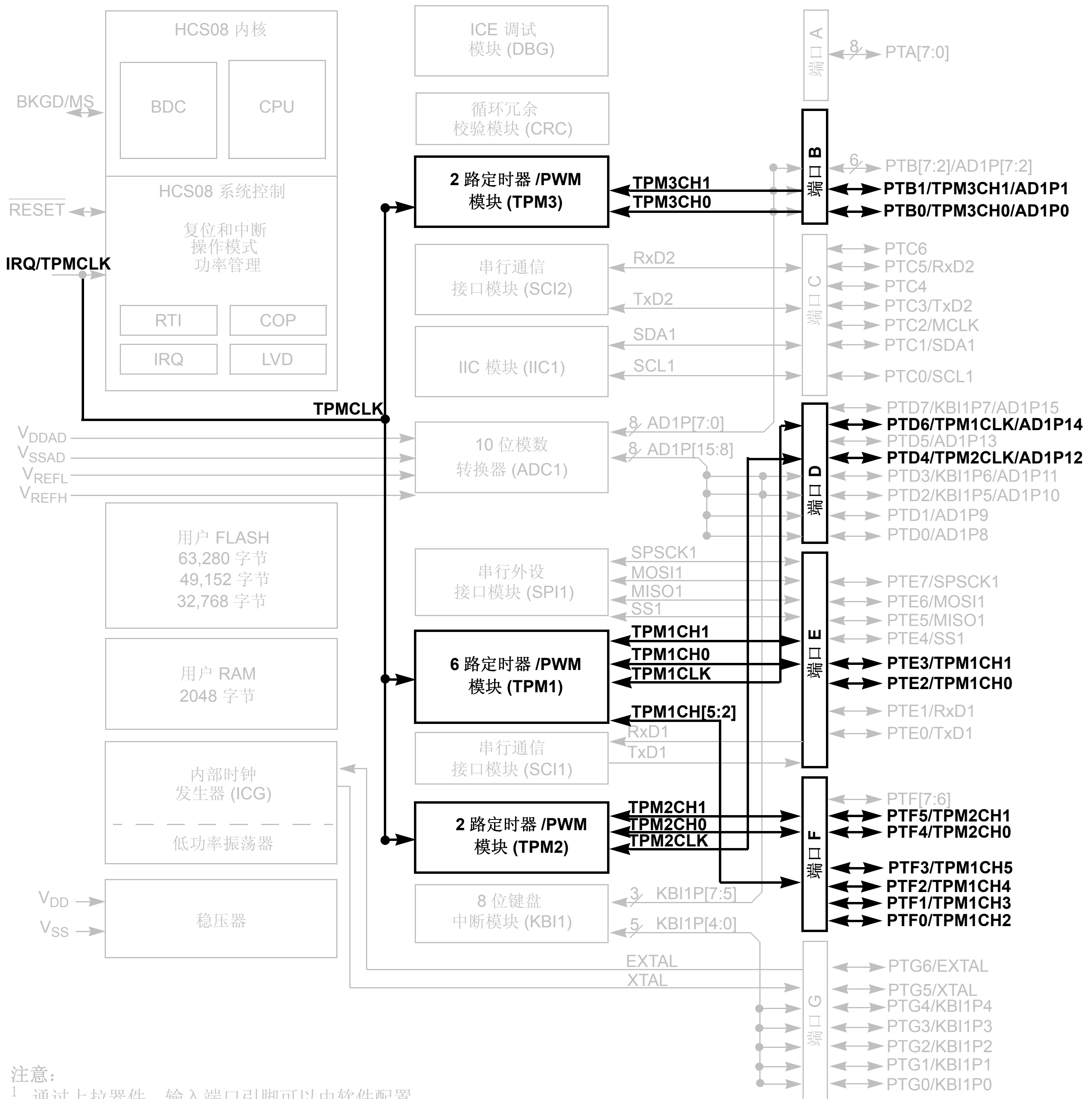
当使用固定系统时钟 （XCLK) 作为任意 TPM 模块的时钟源时，允许 TPM 预分频器用振荡器频率（ICGERCLK/2）除以 2 的频率来运行。只有当 ICG 配置为 FEE 模式并且具备适当条件时，该选项才能使用（参见 第 10 章 内部时钟发生器 (S08ICGV4)）。在所有其他 ICG 模式中，该选项是多余的。因为 XCLK 与 BUSCLK 相同。

外部时钟源能够与 TPMxCLK 引脚连接。总线时钟频率除以 4 就是 TPMxCLK 的最高频率。在 MC9S08AC60 系列中 , TPMCLK、TPM1CLK 和 TPM2CLK 都是通过 SOPT2 寄存器里的 TPMCCFG 位由软件配置。复位后，TPM1CLK、TPM2CLK 和 TPMCLK 分别与 TPM1、TPM2 和 TPM3 连接 （TPMCCFG = 1）。

15.2 功能

定时器系统功能包括：

- 每个 TPM 预分频器时钟源，都可以独立选择总线时钟、固定系统时钟或外部引脚；
- 16 位空转或向下 / 向下 （CPWM）计数操作；
- 16 位模数寄存器控制计数器的范围；
- 定时器系统使能；
- 在每个 TPM 模块上，每通道有一个中断外加一个终端计数中断；
- 各个通道可以是输入捕捉、输出比较或缓冲边缘对齐 PWM ；
- 上升沿、下降沿或任意边沿输入捕捉触发器；
- 设置、清除或转换输出比较操作；
- 在 PWM 输出极性可选；
- 在所有通道上，每个 TPM 都可以配置成缓冲的、中心对齐的脉宽调制 （CPWM)。



注意：
1 通过上拉器件，输入端口引脚可以由软件配置。
2 如果 IRQ 使能 (IRQPE = 1)，引脚包含软件可配置的上拉 / 下拉器件。如果选择上升边沿检测，则下拉使能 (IRQEDG = 1)。
3 引脚包含集成上拉器件。
4 PTD3、PTD2、PTD7 和 PTG4 既包含上拉也包含下拉器件。当 KBI 使能时 (KBIPEn = 1)，且选择了上升边沿 (KBEDGn = 1)，则下拉使能 。
5 TPMCLK、 TPM1CLK 和 TPM2CLK 选项可通过软件配置； 复位后， TPM1CLK、 TPM2CLK 和 TPMCLK 分别提供给 TPM1、 TPM2 和 TPM3。

图 15-1. 显亮 TPM 模块及其引脚的 MC9S08AC60 系列模块图

15.3 TPMV3 与先前版本的差异

TPMV3 是定时器 /PWM 模块的最新版本，它解决了旧版本的一些错误。以下将简要介绍 TPMV3 与 TPMV2 模块差别，以及进行代码移植时的注意事项。

表 15-1. TPMV2 和 TPMV3 代码移植注意事项

操作	TPMV3	TPMV2
写入 **TPMxCnTH:L** 寄存器[1]		
任何写 TPMxCNTH 或 TPMxCNTL 寄存器操作	清除 TPM 计数器 (TPMxCNTH:L) 和预分频器计数器。	只清除 TPM 计数器 (TPMxCNTH:L) 。
读取 **TPMxCNTH:L** 寄存器 [1]		
在 BDM 模式中，任何读取 TPMxCNTH:L 寄存器操作	返回已冻结的 TPM 计数器的值。	在 BDM 模式激活之前，如果只读取了 TPMxCNTH:L 寄存器的一个字节，那就从读缓冲器上返回锁存的 TPMxCNTH:L 值 （而不是冻结的 TPM 计数器值）。
在 BDM 模式中，写 TPMxSC、TPMxCNTH 或 TPMxCNTL 操作。	清除读取一致性机制	不清除读取连贯性机制。
读取 **TPMxCnVH:L** 寄存器[2]		
在 BDM 模式中，任何读 TPMxCnVH:L 寄存器操作。	返回 TPMxCnVH:L 寄存器的值	在 BDM 模式激活之前，如果只读取了 TPMxCnVH:L 寄存器的一个字节，那就从读缓冲器上返回锁存的 TPMxCnTH:L 值 （而不是 TPMxCnVH:L 寄存器中的值）。
在 BDM 模式中，写 TPMxCnSC 操作。	清除读取一致性机制	不清除读取一致性机制。
写入 **TPMxCnVH:L** 寄存器		
在输入捕捉模式中，写 TPMxCnVH:L 寄存器操作[3]	不允许	允许
在输出比较模式中，当 CLKSB:CLKSA 不等于 0:0 时，写 TPMxCnVH:L 寄存器操作 [3]	写入第二个字节以后，在对 TPM 计数器进行下一次修改时 （预分频器计数结束时），用它们的写入缓冲器的值更新 TPMxCnVH:L 寄存器。	只要写入第二个字节就更新这些寄存器。
在边沿对齐 PWM 模式中，当 CLKSB:CLKSA 不等于 00 时，写 TPMxCnVH:L 寄存器操作。	当字节都已经写入，并且 TPM 计数器从 (TPMxMODH:L - 1) 变成 (TPMxMODH:L) 时，用它们的写入缓冲器的值更新 TPMxCnVH:L 寄存器。 **注意 :** 如果 TPM 计数器是空转计数器，那么在 TMP 计数器从 $FFFE 变为 $FFFF 时完成更新。	当 2 个字节都已经写入，并且 TPM 计数器从 TPMxMODH:L 变成 $0000 时进行更新。
在中心对齐 PWM 模式中，当 CLKSB:CLKSA 不等于 00 时，写 TPMxCnVH:L 寄存器操作[4]	当两个字节都已经写入，并且 TPM 计数器从 (TPMxMODH:L - 1) 变为 (TPMxMODH:L) 时，用它们的写入缓冲器的值更新 TPMxCnVH:L 寄存器。 **注意** : 如果 TPM 计数器是空转的计数器，那么当 TPM 计数器从 $FFFE 变为 $FFFF 时便进行更新。	当 2 个字节都已经写入，并且 TPM 计数器从 TPMxMODH:L 变成 (TPMxMODH:L - 1) 时进行更新。
中心对齐 **PWM**		
当 TPMxCnVH:L = TPMxMODH:L[5]	生成 100% 的工作周期。	生成 0% 的工作周期。
当 TPMxCnVH:L = (TPMxMODH:L - 1)[6]	生成接近 100% 的工作周期。	生成 0% 的工作周期。
TPMxCnVH:L 从 0x0000 变成非零值时[7]	等待开始使用新的工作周期设置的新 PWM 周期。	到达当前 PWM 周期的中间时段 （当计数为 0x0000）时，修改通道输出。
TPMxCnVH:L 从一个非零值变成 0x0000[8]	完成使用旧的工作周期设置的当前 PWM 周期。	完成使用新的工作周期设置的当前 PWM 周期。
写入处于 **BDM** 模式的 **TPMxMODH:L** 寄存器		

表 15-1. TPMV2 和 TPMV3 代码移植注意事项 （续）

操作	TPMV3	TPMV2
在 BDM 模式中，写 TPMxSC 寄存器操作	清除 TPMxMODH:L 寄存器的写入一致性机制。	不清除写入一致性机制。

[1] 如需了解更多信息，请参见 15.5.2 节 TPM 计数器寄存器 （TPMxCNTH:TPMxCNHTL）[SE110-TPM 案例 7]
[2] 如需了解更多信息，请参见 15.5.5 节 TPM 通道值寄存器 （TPMxCnVH : TPMxCnVL）
[3] 如需了解更多信息，请参见 15.6.2.1 节 输入捕捉模式 .”
[4] 如需了解更多信息，请参见 15.6.2.4 节 中心对齐 PWM 模式。
[5] 如需了解更多信息，请参见 15.6.2.4 节 中心对齐 PWM 模式 [SE110-TPM 案例 1]
[6] 如需了解更多信息，请参见 15.6.2.4 节 中心对齐 PWM 模式 [SE110-TPM 案例 2]
[7] 如需了解更多信息，请参见 15.6.2.4 节 中心对齐 PWM 模式 [SE110-TPM 案例 3 和 5]
[8] 如需了解更多信息，请参见 15.6.2.4 节 中心对齐 PWM 模式 [SE110-TPM 案例 4]

15.3.1 从 TPMV1 进行移植

移植使用 TPMV1 的器件时，除了考虑 15.3 节 TPMV3 与先前版本的差异外，还需要注意以下事项。

- 当 TPMV2 （而不是 TPMV3）的定时器未处于输入捕捉模式时，您可以写入通道值寄存器（TPMxCnV）。
- 在边沿或中心对齐模式中，只有当定时器从 TPMMOD-1 变为 TPMMOD 时，或者空转定时器从 0xFFFE 变为 0xFFFF 时，通道值寄存器 （TPMxCnV）才会更新。
- 同理，在配置 TPM 模块时，最好在写 TPMxCnV 之前写 TPMxSC，因为写 TPMxSC 会复位 TPMxCnV 寄存器上的连贯性机制。

表 15-2. 移植成 TPMV3 的注意事项

场景	操作 / 最佳实践
写入通道值寄存器 （TPMxCnV） ...	定时器必须为输入捕捉模式。
在边沿或中心对齐模式中，更新通道值寄存器（TPMxCnV） ...	只有当定时器从 TPMMOD-1 变为 TPMMOD （或者当空转定时器 变为 0xFFFF）时才会发生。
将通道值寄存器 （TPMxCnV）的一致性机制复位	写入 TPMxSC。
配置 TPM 模块	首先写入 TPMxSC，然后写入 TPMxCnV 寄存器。

15.3.2 特性

TPM 的主要特性如下：

- 1 到 8 通道：
  - 每个通道可以是输入捕捉，输出比较，边沿对齐 PWM
  - 上升沿、下降沿、或任何边沿输入捕捉触发
  - 置位、清零、输出比较功能
  - PWM 输出极性选择
- 在所有通道上，模块可被设置带缓冲的中心对齐 PWM
- 定时器时钟源可选：分频总线时钟，固定系统时钟，或外部时钟引脚
  - 分频因子可以是 1、 2、 4、 8、 16、 32、 64 或 128
  - 通过片上同步电路固定式系统时钟和总线时钟同步
  - 外部时钟引脚可以与其它任何定时器通道或独立输入引脚共享
- 16 位自运行可调值计数操作
- 定时器系统使能
- 每通道一个中断和计数溢出中断

15.3.3 操作模式

一般来说， TPM 通道可以被独立设置为输入捕捉、输出比较或边沿对齐模式。设置一个控制位，可以使整个 TPM 通道转换为中心对齐 PWM 模式。当使能中心对齐 PWM 模式，在这个 TPM 模块的其它通道上，输入捕捉、输出比较和边沿对齐 PWM 功能将被禁用。

当微控制器处在活跃 BDM 后台调试模式或 BDM 前台调试模式下。 TPM 暂停所有计数直到微控制器返回到正常的用户操作模式。在停止模式下时，所有的系统时钟，包括主要的振荡器，都被停止，因此直到时钟信号重启， TPM 都处于关闭状态。在等待模式下时， TPM 可正常工作。假如 TPM 不需产生一个实时参考时间或提供中断源从等待模式唤醒 MCU，用户可以通过在进入等待模式前禁止 TPM 功能来节省功耗。

- 输入捕捉模式

  当一个设定的边缘事件发生在相关的 MCU 引脚上时， 16 位定时器的当前计数器值被采集到通道值寄存器和置位中断标志位。上升沿，下降沿，任意边沿或无边沿 （禁用通道）可被选作为触发输入比较事件的活跃边沿。

- 输出比较模式

  当定时计数寄存器的值匹配通道值寄存器值时，将置位中断标志位，且相应的 MCU 引脚产生一个可选择输出行为。该输出比较行为可被选来迫使引脚为 0，为 1，锁存它，或忽略该引脚 （用于软件定时功能）。

- 边沿对齐 PWM 模式

  16 位模块寄存器的值加 1 设置 PWM 输出信号的周期。通道值寄存器设置 PWM 输出信号的占空比。用户也可以选择 PWM 输出信号的极性。在该周期末尾和占空比转换临界点，可以产生中断。这种类型的 PWM 信号被称作边沿对齐 PWM 是因为所有 PWM 主要边沿信号与周期开始时对齐，同时一个 TPM 的所有通道都相同。

- 中心对齐 PWM 模式

  16 位模块寄存器值的两倍设置 PWM 输出周期，同时通道值寄存器设置一半占空比的延时。定时器计数递增直到它达到模块值然后计数递减直到 0。当计数值递减并与通道值寄存器相匹配时， PWM 输出活跃。当计数值递增并与通道值寄存器相匹配时， PWM 输出不活跃。这种类型的 PWM 信号被称作

中心对齐 PWM 是因为所有通道活跃占空比周期的中心与计数值零对齐。这种类型的 PWM 被广泛应用于小家电的电机。

这仅是一个概述，在稍后章节将进行详细介绍。

15.3.4 结构框图

TPM 每个通道使用一个输入 / 输出引脚， TPMxCHn （定时器通道 n） n 是通道号 （1-8）。 TPM 输入输出引脚与通用输入输出引脚复用 （请参考相应微控制器的输入输出引脚描述）。图 15-2 是 TPM 结构图。TPM 的主要组成部分是 16 位计数器，可以作为一个自运行计数器或一个递增 / 递减计数器。 TPM 计数器（运行于递增模式时）为输入捕捉，输出比较，边沿对齐 PWM 功能提供参考时钟。定时器计数模块寄存器 TPMxMODH:TPMxMODL 控制计数器因子值 （值 0x0000 或 0xFFFF 可使计数器自运行）。软件可以任意读取计数器值，而不影响计数器的值。 TPMxCNT 计数器的任意一半被写入将复位计数器，而不管写入的是什么。

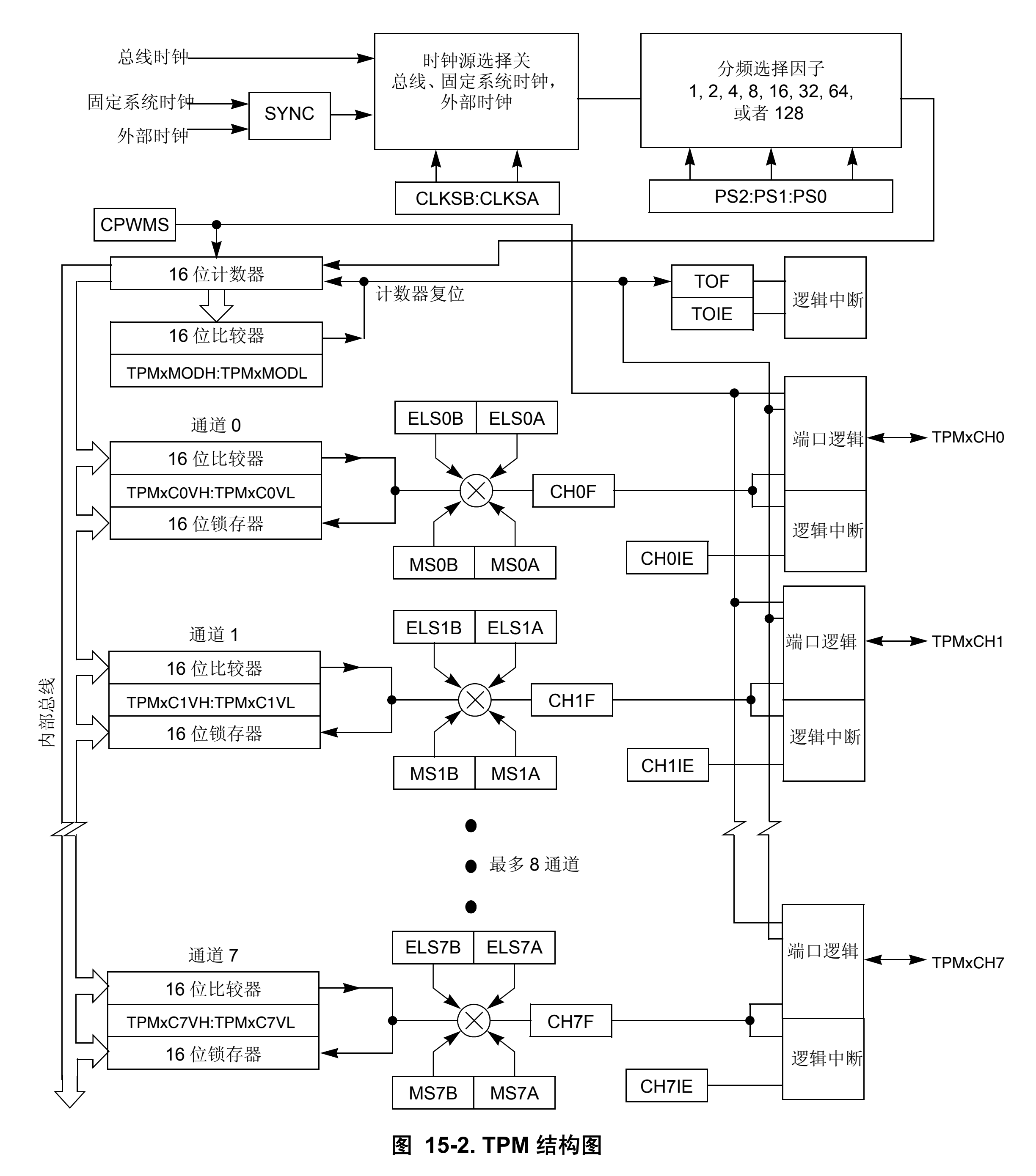


图 15-2. TPM 结构图

TPM 通道和输入捕捉，输出比较或边沿对齐 PWM 通道一样是可独立编程设置的。另一种情况是， TPM 可将所有通道配置成 CPWM 输出。当 TPM 被配置成 CPWM 时，递增计数器 / 递减计数器，输入捕捉，输出比较、 EPWM 等功能无效。

如果通道被配置成输入捕捉模式，则内接一个上拉设备。因为 I/O 口引脚和相关通用 I/O 口控制位不是模块的一部分，模块如何利用引脚相互作用主要取决于芯片的执行情况。可参阅微控制器关于输入输出端口逻辑的描述。

因为中心边沿 PWM 通常被用于驱动 3 相交流电机和直流无刷电机，所以通常使用三通道或六通道。

15.4 信号描述

表 15-3 显示用户可访问的 TPM 信号。通道的数量可以是 1 到 8 个。所有的 TPM 通道都可以复用同一个外部时钟，而且外部时钟只需连接到单独的输入引脚。具体实现可参阅微控制器 I/O 口的描述。

表 15-3. 信号特性

名称	功能
EXTCLK[1]	外部时钟源可被选来驱动 TPM 计数器
TPMxCHn[2]	TPM 通道 n 相关的 I/O 引脚

[1] 若提前预设，该信号可复用给任何通道引脚；然而根据微控制器的实现情况，该信号可被连接到独立的外部引脚。

[2] n = 通道数量 (1 ～ 8)

关于复位状态，端口连接，和这些引脚是否有上拉电阻的详情请参阅该微控制器的说明。 TPM 通道引脚可与通用 I/O 口引脚相关联且有被动上拉电阻，当 TPM 或通用 I/O 控制配置相关引脚为输入时，上拉电阻可以用一个控制位使能。当没有使能 TPM，相应的引脚可应用于 I/O 控制，包括端口数据，数据方向寄存器。复位后，默认没有使能 TPM，所以所有相关引脚被通用 I/O 控制。

15.4.1 信号详细描述

本节将详细描述每个用户可访问的引脚信号。虽然表 15-3 已经列出所有引脚，任何 TPM 引脚可复用外部时钟源信号。因为 I/O 引脚逻辑不是 TPM 的一部分，关于 TPM 引脚功能交互和通用 I/O 控制包括端口数据、数据方向和上拉控制的具体实现可参阅本微控制器的说明。

15.4.1.1 EXTCLK——外部时钟源

位于定时器状态和控制寄存器中的控制位无效 (此时定时器无效)，总线速率时钟 （默认时钟），晶振时钟，或外部时钟可以作为驱动 TPM 分频器和 16 位计数器的时钟。在 TPM 中，外部时钟源是同步的。总线时钟设定时钟同步器；外部时钟频率不能超过总线速率时钟频率的四分之一，满足奈奎斯特准则，并允许抖动。

外部时钟信号和通道 I/O 引脚复用相同引脚。所以当被选做外部时钟源时，引脚通道 I/O 功能不可用。用户应避免产生上述情况。如果这个引脚可用作外部时钟源 （CLKSB:CLKSA），该通道能在输出比较模式下用作软件定时器 （ELSnB:ELSnA=0:0）。

15.4.1.2 TPMxCHn——TPM 通道 n 输入输出引脚

MCU 的每个 TPM 通道与一个 I/O 引脚相关联。引脚功能依赖于通道配置。 TPM 引脚与通用 I/O 引脚复用，它的每个引脚对应一个端口数据寄存器位，一个数据方向控制位，并且作为输入时，端口具有可使能的被动上拉电阻。

当 （ELSnB:ELSnA = 0:0）或 （CLKSB:CLKSA = 0:0）时， TPM 通道不能控制 I/O 引脚。所以通常转为通用 I/O 控制。当 CPWMS = 1 （ELSnB:ELSnA ≠ 0:0）时，所有带有 TPM 的通道被配置成中心对齐 PWM 且 TPMxCHn 引脚被 TPM 控制。当 CPWMS = 0 时， MSnB: MSnA 控制位决定该通道配置为输入捕捉，输出比较或边沿对齐 PWM。

当一个通道被配置为输入捕捉模式 （CPWMS = 0， MSnB:MsnA = 0:0， ELSnB:ELSnA ≠ 0:0）时， TPMxCHn 引脚被作为 TPM 的边沿有效输入。 ELSnB:ELSnA 控制位决定触发输入捕捉事件的极性边沿或者边沿。基于总线时钟的同步器用于将输入边沿同步到总线时钟上。这意味着在一个输入捕捉引脚上能被可靠检

测到的最小脉冲宽度是总线时钟周期的四倍 （能检测接近两个总线时钟的理想时钟脉宽）。 TPM 把这个引脚当作输入捕捉的输入，而忽略端口数据和数据方向控制。

当通道被配置为输出比较模式（CPWMS = 0，MSnB: MsnA = 0:1，ELSnB:ELSnA = 0:0）时，忽略相关数据方向控制， TPMxCHn 引脚被认为是由 TPM 控制输出， ELSnB:ELSnA 控制位决定该引脚如何被控制。ELSnB:ELSnA 剩余的三种组合决定 TPMxCHn 引脚是否被锁存，清空或每次 16 位通道值寄存器匹配定时器计数器时置位。当输出比较模式被选中，引脚上的前值被清除直到下个输出比较事件来临然后引脚被锁存。

当一个通道被配置为边沿对齐 PWM 模式（CPWMS = 0，MSnB = 1，ELSnB:ELSnA ≠ 0:0）时，数据方向被忽略， TPMxCHn 引脚被强制为由 TPM 控制输出， ELSnA 控制引脚上 PWM 输出信号的极性。当 ELSnB:ELSnA = 1:0 时， TPMxCHn 引脚在每个新周期 （TPMxCNT = 0x0000）开始被强制拉高，当通道值寄存器等于定时器的计数器时引脚被强制拉低。当 ELSnA = 1 时，TPMxCHn 引脚在每个新周期（TPMxCNT = 0x0000）开始被强制拉低，当通道寄值存器等于定时器的计数器时引脚被强制拉高。

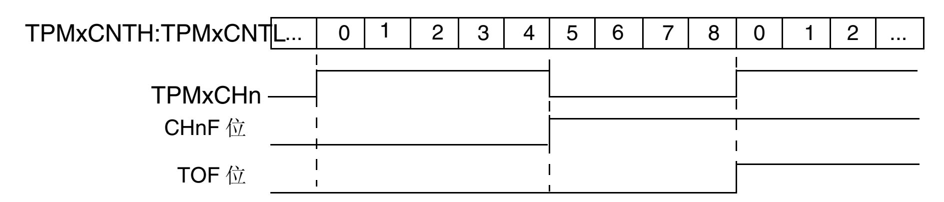


图 15-3. 边缘对齐 PWM 模式的高真脉冲

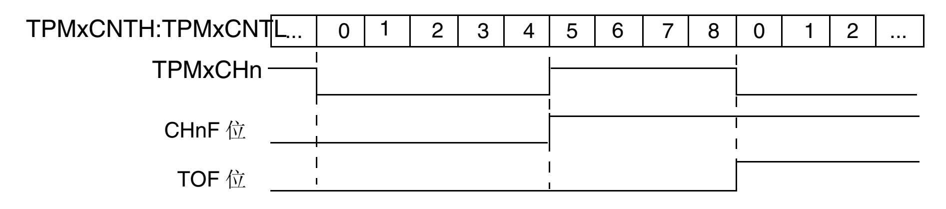


图 15-4. 边缘对齐 PWM 模式的低真脉冲

当 TPM 被配置为中心对齐 PWM 模式 （ELSnB:ELSnA ≠ 0:0），在 TPM 模式下，所有通道的数据方向被忽略， TPMxCHn 引脚被强制转换为由 TPM 控制输出， ELSnA 位控制每个 TPMxCHn 输出的极性。如果 ELSnB:ELSnA = 1:0，当定时器计数器值上升，通道值寄存器等于定时器计数器时 TPMxCHn 引脚被清零；当定时器计数器值下降，通道值寄存器等于定时器计数器值时， TPMxCHn 引脚被置位。如果 ELSnA = 1，当定时器计数器值上升且通道值寄存器等于定时器计数器时， TPMxCHn 引脚被置位；当定时器计数器值下降且通道值寄存器等于定时器计数器时 TPMxCHn 引脚被清零。

TPMxMODH:TPMxMODL = 0x0008
TPMxMODH:TPMxMODL = 0x0005

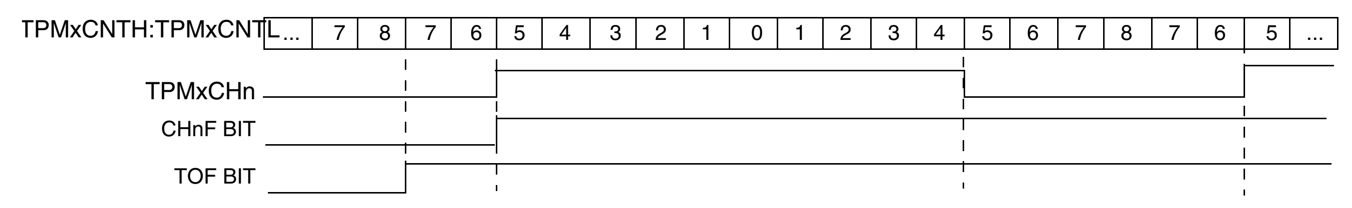


图 15-5. 中心对齐 PWM 模式的高真脉冲

TPMxMODH:TPMxMODL = 0x0008
TPMxMODH:TPMxMODL = 0x0005

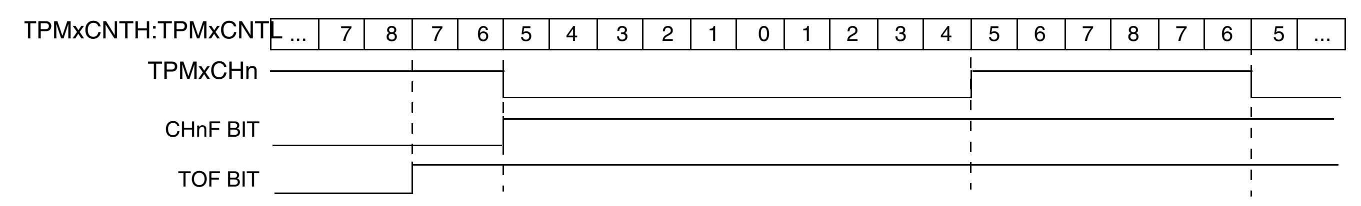


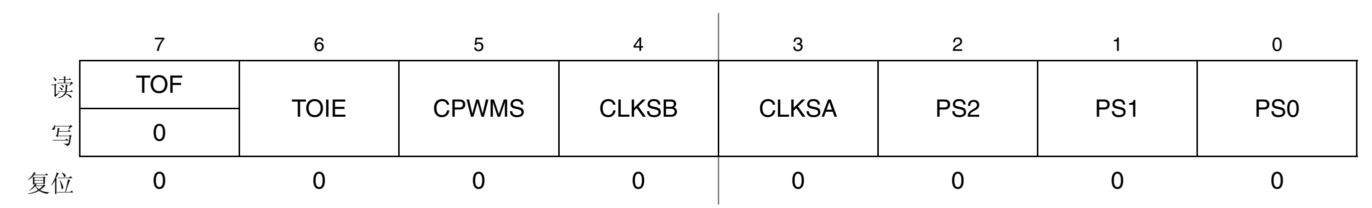
图 15-6. 中心对齐 PWM 模式的低真脉冲

15.5 寄存器定义

本节将按地址顺序描述寄存器。一个典型 MCU 可能包含若干 TPM，每个 TPM 有八个通道，所以寄存器的命名包含占位符字用来明确哪个 TPM 和哪个通道被选用了。例如 TPMxCnSC 涉及到 (TPM)x，通道 n。TPM1C2SC 可能是定时器 1 通道 2 的状态和控制寄存器。

15.5.1 TPM 状态和控制寄存器 （TPMxSC）

TPMxSC 包含溢出状态标记和用来配置中断使能、 TPM 配置、时钟源、分频因子的控制位。这些控制涉及到此定时器模块的所有通道。



	7	6	5	4	3	2	1	0
读	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
写	0							
复位	0	0	0	0	0	0	0	0

图 15-7. TPM 状态和控制寄存器 （TPMxSC）

表 15-4. TPMxSC 位描述

位	描述
7 TOF	定时器溢出标志，当 TPM 计数器达到在 TPM 计数器模寄存器中预设值后 , 复位为 0x0000, 然后置位这个读写标志。当 TOF 被置位，可以通过读 TPM 状态和控制寄存器，然后向 TOF 写 0 清除 TOF。如果在彻底清除序列前发生了另一个 TPM 溢出，该序列复位，并在先前清除序列完成后， TOF 还将保持置位。这样在清序列时 TOF 中断请求不会丢失。复位清零 TOF。对 TOF 写 1 无效。 0 TPM 计数器未达到模值或未溢出 1 TPM 计数器溢出
6 TOIE	定时器溢出中断使能。这个读写位使能 TPM 溢出中断。如果 TOIE 被置位，当 TOF 等于 1 中断产生。复位清除 TOIE。 0 TOF 中断禁止 （用于软件轮询） 1 TOF 中断使能
5 CPWMS	中心对齐 PWM 选择位。这个读写位选择 CPWM 操作模式。默认情况下， TPM 提供输入捕捉，输出比较和边沿对齐 PWM 功能，并提供递增计数器操作。设置 CPWMS 重新配置 TPM 操作于上升 / 下降模式计数器，实现 CPWM 功能。复位清零 CPWMS。 0 所有通道可作输入捕捉，输出比较，边沿对齐 PWM 模式，由 MSnB: MsnA 控制每个通道的状态和控制寄存器中的相应位来选择 1 所有通道都可运行在中心对齐 PWM 模式下
4-3 CLKS[B:A]	时钟源选择位，如表 15-5 所示，这 2 个位用来禁止 TPM 系统或选择三个时钟源中的一个来驱动计数器分频器。固定系统时钟源在带有一个基于系统 PLL 时钟的系统中是唯一有效的。若没有 PLL，固定系统时钟与总线速率时钟相同。外部时钟源通过 TPM 模块同步总线时钟，固定系统时钟源通过片上同步电路同步总线时钟。当有 PLL 但无效时，固定系统时钟与总线速率时钟相同。
2-0 PS[2:0]	分频因子选择位。这三位选择 8 个分频因子中的一个用来分频 TPM 时钟输入，如表 15-6 所示。在任意时钟源同步或时钟源选择后，分频器被定位，所以它影响用来驱动 TPM 系统的时钟源。在新值被更新到寄存器后，新的分频因子将对下个系统时钟周期产生影响。

表 15-5. TPM 时钟源选择

CLKSB:CLKSA	TPM 时钟源分频输入
00	无时钟选择
01	总线速率时钟
10	固定系统时钟
11	外部时钟源

表 15-6. 分频因子选择

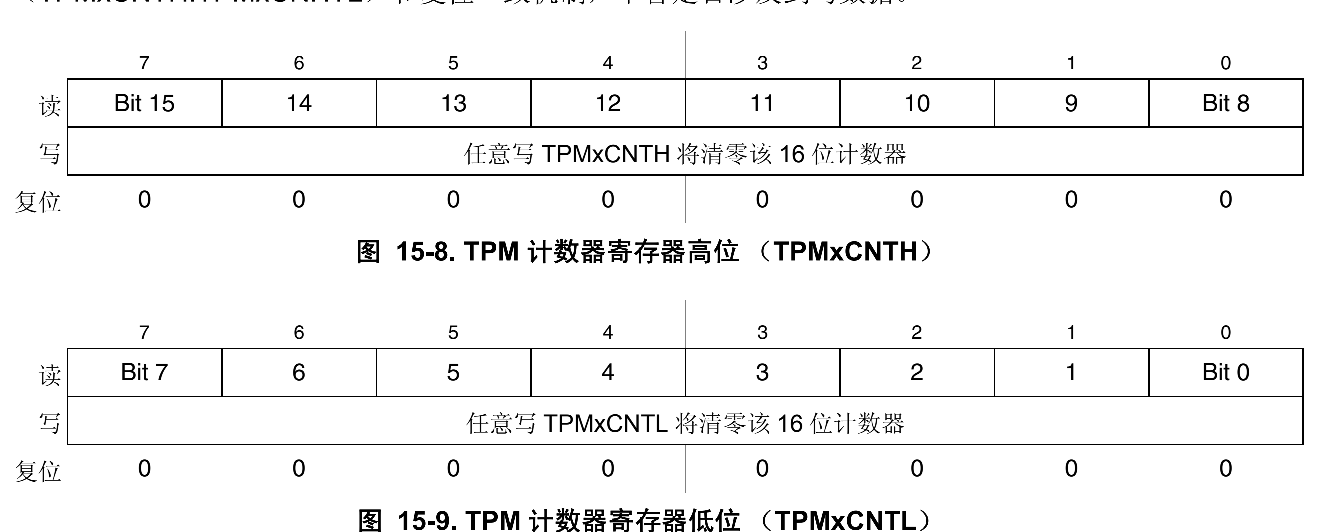
PS2:PS1:PS0	分频因子
000	1
001	2
010	4
011	8

表 15-6. 分频因子选择 (续)

PS2:PS1:PS0	分频因子
110	16
101	32
110	64
111	128

15.5.2 TPM 计数器寄存器 （TPMxCNTH:TPMxCNHTL）

这两个只读 TPM 计数器寄存器包含其高位和低位字节内容。在读 TPMxCNTH 或 TPMxCNHTL 中的一个字节时，两字节内容都被锁存到一个缓冲区，直到剩下的一个字节内容也被读出。这使得 16 位连读而更方便于各种编译器编译。这种一致机制是通过一个 MCU 复位或任意写定时器状态和控制寄存器 （TPMxSC）自动重新开始的。复位清空 TPM 计数器寄存器。写任意值到 TPMxCNTH 或 TPMxCNHTL 同样清空 TPM 计数器 （TPMxCNTH:TPMxCNHTL）和复位一致机制，不管是否涉及到写数据。



	7	6	5	4	3	2	1	0
读	Bit 15	14	13	12	11	10	9	Bit 8
写	任意写 TPMxCNTH 将清零该 16 位计数器							
复位	0	0	0	0	0	0	0	0

图 15-8. TPM 计数器寄存器高位 （TPMxCNTH）

	7	6	5	4	3	2	1	0
读	Bit 7	6	5	4	3	2	1	Bit 0
写	任意写 TPMxCNTL 将清零该 16 位计数器							
复位	0	0	0	0	0	0	0	0

图 15-9. TPM 计数器寄存器低位 （TPMxCNTL）

当 BDM 被激活，定时器计数器被暂停 （这个值将被用户读取）；当 BDM 被激活时，一致机制被暂停，缓冲区保持在原来的状态，即使计数器的部分或全部内容在 BDM 活跃时被读取。这保证了当 BDM 被激活时如果用户正在读 16 位寄存器时，在返回到正常执行状态下它将会读 16 位的其它相关值。

在 BDM 模式下，写任意值到 TPMxSC、 TPMxCNTH、 TPMxCNTL 寄存器将复位 TPMxCNTH:L 寄存器读一致机制，不管是否涉及写的数据。

15.5.3 TPM 计数器模寄存器 （TPMxMODH:TPMxMODL）

该读写 TPM 模寄存器包含其模值。当 TPM 计数器达到模值之后， TPM 计数器在下个时钟重新开始从 0X0000 计数，溢出标志 （TOF）被置位。写 TPMxMODH 或 TPMxMODL 禁止 TOF 位和溢出中断直到其他字节被写完。复位设置 TPM 计数器模寄存器为 0X0000，由一个自运行定时器计数器产生 （模块禁止）。

写 TPMxMODH:TPMxMODL 其中的一个字节锁存该值到一个缓冲区里，同时寄存器被依据 CLKSB:CLKSA 位值决定的写缓冲区值所更新，所以：

- 如果 （CLKSB:CLKSA = 0:0），则寄存器在第二个字节被写后更新

- 如果 （CLKSB:CLKSA!=0:0）则寄存器在两个字节被写后更新， TPM 计数器从 （TPMxMODH:TPMxMODL - 1）改变为 （TPMxMODH:TPMxMODL）。如果 TPM 计数器是一个自运行计数器，则更新发生在 TPM 计数器值从 0XFFFE 变为 0XFFFF。

该闭锁机制可通过写 TPMxSC 地址 （不论 BDM 是否活跃）手动复位。

当 BDM 活跃时，一致机制被暂停 （除非通过写 TPMxSC 寄存器来复位），即使模寄存器的一半或者全部在 BDM 活跃时被写，缓冲锁仍保持在 BDM 激活时的状态。当 BDM 活跃时，任何写模块寄存器将越过缓冲锁存而直接写模块寄存器。



	7	6	5	4	3	2	1	0
读 写	位 15	14	13	12	11	10	9	位 8
复位	0	0	0	0	0	0	0	0

图 15-10. TPM 计数器模寄存器高位 （TPMxMODH）

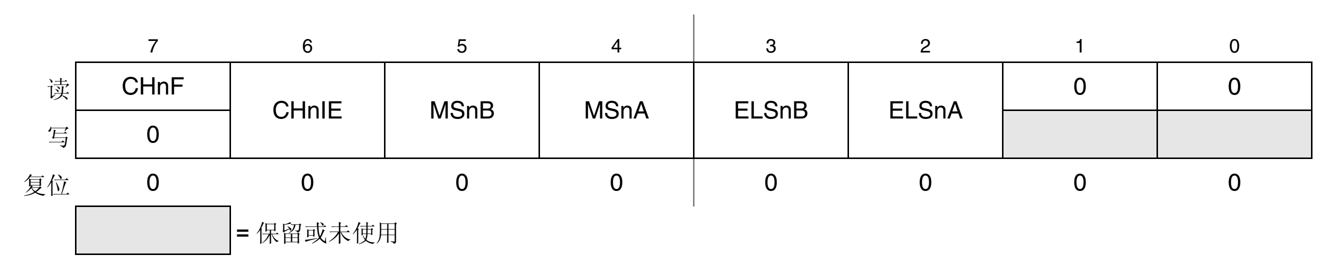
	7	6	5	4	3	2	1	0
读 写	位 7	6	5	4	3	2	1	位 0
复位	0	0	0	0	0	0	0	0

图 15-11. TPM 计数器模寄存器低位 （TPMxMODL）

在写 TPM 模寄存器前复位 TPM 计数器，以避免混淆计数器第一次溢出发生的时间。

15.5.4 TPM 通道 n 状态和控制寄存器 （TPMxCnSC）

TPMxCnSC 包含通道中断状态标志和控制位用来配置中断使能，通道配置和引脚功能。



	7	6	5	4	3	2	1	0
读	CHnF	CHnIE	MSnB	MSnA	ELSnB	ELSnA	0	0
写	0							
复位	0	0	0	0	0	0	0	0

= 保留或未使用

图 15-12. TPM 通道 n 状态和控制寄存器 （TPMxCnSC）

表 15-7. TPMxCnSC 位描述

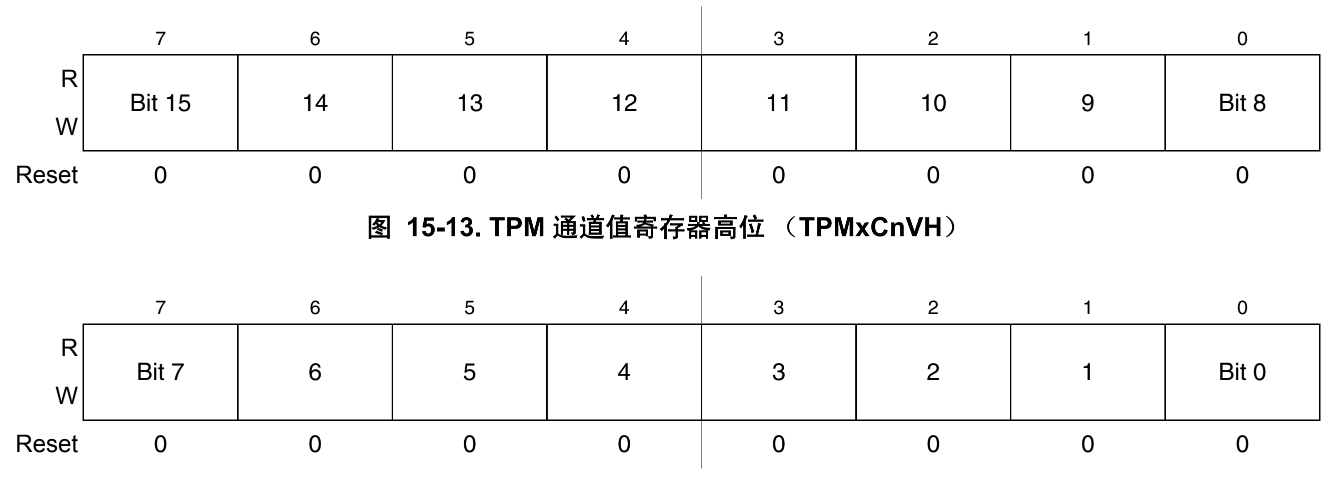
位	描述
7 CHnF	通道 n 标志。 当通道 n 为输入捕捉通道时，通道 n 上的有效边沿触发该位置位。当通道 n 为输出比较或边沿 / 中心对齐 PWM 通道时，当 TPM 计数器的值等于 TPM 通道 n 寄存器的值时 CHnF 被置位。当通道 n 为边沿对齐 / 中心对齐 PWM 通道且占空比被设为 0% 或 100% 时，即使当 TPM 计数器的值等于 TPM 通道 n 寄存器的值时， CHnF 也不会被置位。当 CHnF 被置位且中断允许 （ChnIE=1）时，产生相应的中断请求。当 CHnF 被置位时通过读 TPMxCnSC，并向 CHnF 写 0 清空 CHnF。在清空序列完成前如果另一个中断请求发生，该序列复位，并且在清空前一个中断序列之后， CHnF 仍保持置位。这样，不会因为清除前一个中断标志，而丢失新的 CHnF 中断请求。复位清空 CHnF 位。向 CHnF 写 1 不产生影响。 0 无输入捕捉或输出比较事件发生在通道 n 上 1 输入捕捉或输出比较事件发生在通道 n 上
6 ChnIE	通道 n 中断使能位。该读写位使能通道 n 的中断。复位清空 ChnIE。 0 禁止通道 n 中断请求 1 允许通道 n 中断请求
5 MSnB	TPM 通道 n 的 B 模式。当 CPWMS = 0，MSnB = 1 时配置 TPM 通道 n 为边沿对齐 PWM 模式。参阅表 15-8 中通道模式和设置控制。
4 MSnA	TPM 通道 n 的 A 模式。当 CPWMS=0， MSnB=0 时， MSnA 配置 TPM 通道 n 为输入捕捉模式或输出比较模式。参阅表 15-8 中通道模式和设置控制。 注：在转变为输入捕捉模式之前，如果相关端口引脚在至少两个总线时钟周期内不稳定，将会出现意外的边沿触发。
3-2 ELSnB ELSnA	边沿 / 电平选择位。 取决于定时器通道的操作模式，通过 CPWMS:MSnB:MSnA 来设置，详细可参照表 15-8。这些位选择输入边沿的极性来触发一个输入捕捉事件，选择电平驱动一个输出比较匹配或选择 PWM 输出极性。通过设置 ELSnB:ELSnA = 0:0 配置相关定时器引脚作为通用 I/O 引脚。这种功能用于暂时禁止输入捕捉，或当相关定时器通道被设为不需使用引脚的软定时器时 , 使定时器引脚作通用 I/O 脚。

表 15-8. 模式、边沿和电位选择

CPWMS	MSnB:MsnA	ELSnB:ELSnA	模式	配置
X	XX	00	引脚不用于 TPM 转换为通用 I/O 或其它外围控制	
CPWMS	MSnB:MsnA	ELSnB:ELSnA	模式	配置
0	00	01	输入捕捉	仅在上升沿捕捉
		10		仅在下降沿捕捉
		11		上升沿或下降沿捕捉
	01	01	输出比较	比较时锁存输出
		10		比较时清空输出
		11		比较时设置输出
	1X	10	边沿对齐 PWM	高保真脉宽 （比较时清空输出）
		X1		低保真脉宽 （比较时设置输出）
1	XX	10	中心对齐 PWM	高保真脉宽 （比较时清空输出）
		X1		低保真脉宽 （比较时设置输出）

15.5.5 TPM 通道值寄存器 （TPMxCnVH : TPMxCnVL）

这些读写寄存器包含通过输入捕捉功能被捕获的 TPM 计数器值或输出比较的值或 PWM 功能。复位清空通道寄存器。



	7	6	5	4	3	2	1	0
R W	Bit 15	14	13	12	11	10	9	Bit 8
Reset	0	0	0	0	0	0	0	0

图 15-13. TPM 通道值寄存器高位 （TPMxCnVH）

	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
Reset	0	0	0	0	0	0	0	0

图 15-14. TPM 通道值寄存器低位 （TPMxCnVL）

在输入捕捉模式下，在读 （TPMxCnVH : TPMxCnVL）任一字节过程中，寄存器两个字节的内容将被锁存到缓冲器中，直至读取另外一个字节。当 TPMxCnSC 寄存器被写 （BDM 模式是否活跃）时，这个锁机制可被复位 （变为非锁定）。任何对通道寄存器的写操作在输入捕捉模式下将被忽略。当 BDM 活跃时，该一致机制被暂停 （除非通过写 TPMxCnSC 寄存器复位），这样，即使读部分或全部通道寄存器，缓冲器锁存值保持在 BDM 变为活跃时的状态。这确保了正在读 16 位寄存器时，用户使能 BDM，而在返回到正常执行后，仍能正确读取 16 位值的另一半。在 BDM 模式下从 TPMxCnVH : TPMxCnVL 寄存器读取的值是寄存器的值而不是缓冲区的值。

在输出比较或 PWM 模式下，写 （TPMxCnVH : TPMxCnVL）其中的一个字节，并锁存到缓冲区。在两个都被写完后，根据 CLKSB:CLKSA 位的值和所选的模式被转换成一个 16 位值放入到定时器通道寄存器，所以：

- 如果 CLKSB:CLKSA=0:0，当第二个字节被写入时寄存器被更新。
- 如果 CLKSB:CLKSA!=0:0 且在输出比较模式下，在第二个字节被写后，并且在 TPM 计数器下个变化时 （分频计数结束）寄存器被更新。
- 如果 CLKSB:CLKSA!=0:0 并且在 EPWM 或 CPWM 模式下，在两个字节被写后， TPM 计数器从（TPMxMODH : TPMxMODL - 1）变为 （TPMxMODH : TPMxMODL），寄存器被更新。如果 TPM 计数器是自运行计数器，则当 TPM 计数器从 0XFFFE 变为 0XFFFF 时发生更新操作。

该锁机制可以通过写 TPMxCnSC 寄存器 （不论 BDM 是否活跃）进行手动复位。该锁机制允许以升序或降序写入 16 位值，方便各种编译器执行。

当 BDM 活跃，该一致机制被暂停 （除非通过写 TPMxCnSC 寄存器复位），该缓冲锁存保持在原状态，即使部分或全部通道寄存器被写。当 BDM 活跃时，任何写通道寄存器将越过缓冲锁存而直接写通道寄存器。在 BDM 活跃时且恢复正常执行时，写入通道寄存器的值被用于 PWM 和输出比较操作。当 BDM 活跃，写通道寄存器不需一致序列的干预。在一致机制被完全实行，用用户写入缓冲的值 （BDM 不活跃时）来更新通道寄存器。

15.6 功能描述

所有的 TPM 功能与 16 位计数器相关，此计数器允许灵活的选择时钟源和分频因子。同时还有一个 16 位模寄存器相应于此主计数器。

CPWMS 控制位为 TPM 的所有通道选择中心对齐 PWM （CPWMS=1）或通用定时功能（CPWMS=0），该功能可以独立配置使之工作在输入捕捉、输出比较或边沿对齐 PWM 模式下。该 CPWMS 控制位位于主 TPM 状态和控制寄存器，因为它对影响 TPM 的所有通道，并决定主计数器的操作方式。（在 CPWM 模式下，作为通用计数器功能计数器通常为增 / 减模式）。

下面各节将描述核心计数器和每个定时器的操作模式 （输入捕捉，输出比较，边沿对齐 PWM，中心对齐 PWM）。因为引脚操作的细节和中断活跃取决于操作模式，相应的内容在相关模式章节进行讲述。

15.6.1 计数器

所有定时器功能建立在核心 16 位计数器基础之上 （TPMxCNTH : TPMxCNTL）。本节将讨论时钟源的选择、计数结束溢出、计数增 / 减、计数器复位。

15.6.1.1 计数器时钟源

CLKSB:CLKSA 位于定时器状态和控制寄存器 （TPMxSC），用来选择三个时钟源中的一个或 OFF （能有效禁止 TPM）。见表 15-5。MCU 在复位后，CLKSB:CLKSA = 0:0，没有时钟源被选择，TPM 处于低功耗状态。这些控制位可在任意时间被读或写，禁止定时器 （写 00 到 CLKSB:CLKSA）不影响计数器或其它定时器寄存器中的值。

总线速率时钟是 MCU 的主要系统总线时钟。该时钟源无需同步，因为该时钟用于所有内部 MCU 活动包括 CPU 和总线操作。

在不带 PLL 电路的 MCU 中或不使用 PLL 电路时，固定系统时钟源和和总线速率时钟源相同，不需要使用同步器。当使用 PLL 或者 FLL 电路时，在二分频时钟源和定时器计数器之间需要一个同步装置，以保证计数器转换和总线时钟转换相一致。一个同步装置将被用在芯片层，使与晶振相关的时钟和总线时钟同步。

外部时钟可被接到 TPM 通道的任意脚。该时钟源通过一个同步装置来确保计数器转换和总线时钟转换能一致。总线速率时钟驱动同步装置；为满足奈奎斯特定理 （Nyquist）采集信号的要求，外部时钟源的频率不得比总线速率时钟的四分之一更快。理想情况下外部时钟等于总线时钟除以 4。

当外部时钟源复用 TPM 通道引脚时，该引脚不能被用于其它通道的定时功能。例如，当 TPM 通道 0 引脚被用作定时器外部时钟源时，不能再把它配置成输入捕捉 （用户应尽量避免）。 TPM 通道仍然能在输出比较模式中使用软件定时功能 （引脚控制设置不影响 TPM 通道引脚）。

15.6.1.2 计数器溢出和模复位

一个中断标志和中断使能与 16 位核心计数器相关。软件可访问的 TOF 用来标识定时器计数器溢出。该使能信号可选择软件轮询 （TOIE = 0，即无硬件中断产生）或中断驱动 （TOIE = 1，即一个硬件中断产生）。

TOF 是否置位取决于 TPM 是否配置为中心对齐 PWM 模式 （CPWMS=1）。在简单模式下，没有模数限制， TPM 不是中心对齐 PWM 模式。在这种情况下，该 16 位计数器从 0x0000 计数到 0xFFFF，在下个计数时钟溢出为 0X0000。当从 0xFFF 变为 0x000 时 TOF 置位。当 TPM 处于中心对齐 PWM 模式时（CPWMS = 1），当计数器达到模数寄存器中到预设值，并改变计数方向时， TOF 置位。这依据于 PWM 周期结束时的值。（0x000 计数值对应于一个周期的中间值）。

15.6.1.3 计数模式

该核心定时器计数器有两种计数模式。当选择中心对齐 PWM 模式时 （CPWMS = 1），计数器在增 / 减模式下操作。在其它情况下，计数器用作单增计数器。作为单增计数器，定时器计数器从 0x0000 计数直到终点值然后再从 0x0000 重新开始计数。该终端计数是 0xFFFF 或是 TPMxMODH : TPMxMODL 设定的模数值。

当被指定为中心对齐 PWM 模式操作时，定时器计数器从 0x0000 计数直到终点值然后再下降到 0x0000 循环计数。 0x0000 和终点计数值都是正常长度计数值 （一个定时器周期长）。在该模式下，定时器溢出标志（TOF）在终点计数周期结束时 （计数变为下个更小的计数值时）被置位。

15.6.1.4 手动计数器复位

在任何时候核心定时器计数器可通过写任意值到 TPMxCNTH : TPMxCNTL 其中之一来手动复位。用这种方式复位计数器同时也复位了一致机制，以防止在复位计数前仅读取部分计数器值。

15.6.2 通道模式选择

若 CPWMS=0，通道 n 状态和控制寄存器中的 MSnB 和 MSnA 控制位决定相关通道操作的基本模式。包含输入捕捉，输出比较和边沿对齐 PWM。

15.6.2.1 输入捕捉模式

在输入捕捉模式中， TPM 能够捕捉一个外部事件发生的时间。当一个活跃边沿发生在输入捕捉通道引脚上时， TPM 锁存其计数器内容到通道值寄存器 （TPMxCnVH : TPMxCnVL）。上升沿、下降沿或任意边沿可被选作活跃边沿触发一个输入捕捉。

在输入捕捉模式下， TPMxCNTH : 和 TPMxCNTL 寄存器只读。

当 16 位捕捉寄存器任一半被读时，另一半被锁存到缓冲中来保证 16 位被一致存取。该一致序列可通过写通道状态和控制寄存器 （TPMxCnSC）来手动复位。

一个输入捕捉事件设置一个标志位 （CHnF）可产生一个 CPU 中断请求。

在 BDM 模式时，输入捕捉功能可由用户来配置。当一个外部事件发生时， TPM 锁存其计数器内容 （在 BDM 模式下被暂停计数）到通道值寄存器并设置标志位。

15.6.2.2 输出比较模式

有输出比较功能时， TPM 可以根据预设的位置、极性、周期、频率来产生定时脉冲。当计数器达到输出比较通道的通道值寄存器中设置的值时，该 TPM 可置位、清空或锁存通道引脚。

在输出比较模式下，仅在 16 位寄存器的两个 8 位都被写后，参考 CLKSB: CLKSA 位的值，数值被传递到相应定时器通道寄存器。所以：

- 如果 （CLKSB: CLKSA=0:0），当写入第二个字节，寄存器被更新
- 如果 （CLKSB: CLKSA!=0:0），第二个字节被写后，在 TPM 计数器下个改变时寄存器被更新。

该一致序列可通过写通道状态和控制寄存器 （TPMxCnSC）来手动复位。

一个输出比较事件设置一个标志位 （CHnF），可产生一个 CPU 中断请求。

15.6.2.3 边沿对齐 PWM 模式

这种 PWM 输出类型使用定时器计数器单增计数模式 （CPWMS=0）且当 TPM 中的其它通道被配置为输入捕捉或输出比较功能时被使用。该 PWM 信号周期由模寄存器值 （TPMxMODH : TPMxMODL）加 1 决定。占空比由设置在定时器通道寄存器 （TPMxCnVH : TPMxCnVL）的值来决定。该 PWM 信号的极性由 ELSnA 控制位来决定。 1％和 100％占空比是可能的。

在 TPM 通道值寄存器中的输出比较值决定 PWM 信号的脉宽 （占空比）（图 15-15）。模数溢出和输出比较之间的时间为脉冲宽度。如果 ELSnA = 0，计数器溢出强制 PWM 信号为高，输出比较强制 PWM 信号为低。如果 ELSnA = 1，计数器溢出强制 PWM 信号为低，输出比较强制 PWM 信号为高。

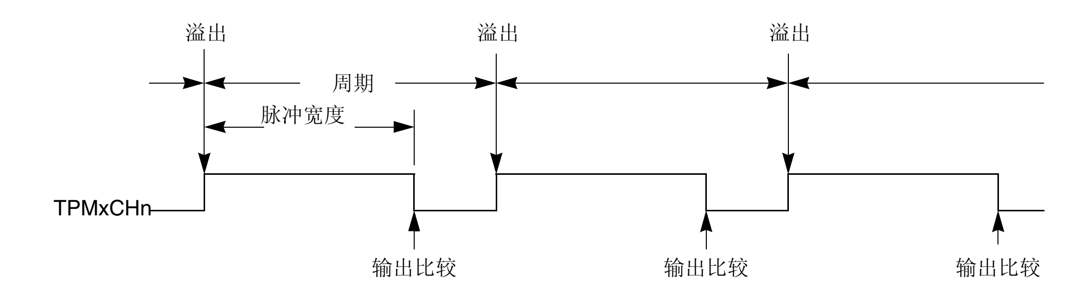


图 15-15. PWM 周期和脉冲宽度 （ELSnA= 0）

当通道值寄存器被设为 0x0000 时，占空比为 0%。通过设置定时器寄存器 （TPMxCnVH : TPMxCnVL）为一个大于模数设置的值，可达到 100% 占空比。这表明为了获得 100% 占空比模数设置必须小于 0xFFFF。

因为 TPM 可用于 8 位 MCU，定时器通道寄存器中的设置被放到缓冲以确保 16 位一致更新和避免不可预料的脉宽。写 TPMxCnVH 和 TPMxCnVL 寄存器实际上是写缓冲寄存器。在边沿对齐 PWM 模式下，参考 CLKSB: CLKSA 位的值，数值可以传递到相应定时器通道寄存器。所以：

- 如果 （CLKSB:CLKSA = 0:0），当写第二个字节时，寄存器被更新
- 如果 （CLKSB:CLKSA != 0:0），在两个字节都被写入后， TPM 计数器从 （TPMxMODH : TPMxMODL -1）变为 （TPMxMODH : TPMxMODL），寄存器被更新。如果 TPM 计数器是一个自运行计数器则当 TPM 计数器从 0XFFFE 变为 0XFFFF 时，更新被执行。

15.6.2.4 中心对齐 PWM 模式

这种 PWM 输出类型使用定时器计数器增 / 减计数模式 （CPWMS = 1）。当周期是由 TPMxMODH TPMxMODL 中的值决定时， TPMxCnVH : TPMxCnVL 中的输出比较值决定了 PWM 信号的脉宽 （占空比）。TPMxMODH:TPMxMODL 应该被保持在 0x0000 到 0x7FFF 范围之间，因为这个范围之外的值易产生不确定结果。 ELSnA 决定 CPWM 输出的极性。

脉宽 = 2 × （TPMxCnVH : TPMxCnVL）

周期 = 2 × （TPMxMODH:TPMxMODL）； TPMxMODH:TPMxMODL = 0x0001- 0x7FFF

如果通道值寄存器 TPMxCnVH : TPMxCnVL 为零或负数 （位 15 置位）， 占空比将为 0%。如果 TPMxCnVH : TPMxCnVL 为正数 （位 15 清空）且大于模数设定值 （非负），占空比为 100%，这是因为绝不会发生占空比相比。这表明由模数寄存器设定的可用周期范围是从 0x0001 到 0x7FFE。这不是一个重要的限制。产生结果的周期将比一般应用的周期长很多。

TPMxMODH:TPMxMODL = 0x0000 是一个特殊情况，通常不用于中心对齐 PWM 模式。当 CPWMS = 0 时，计数器将从 0x0000 到 0xFFFF 自动运行，但当 CPWMS = 1 时，为了改变计数方向从单增计数转为单减计数，该计数器需要在大于 0x0000 的某处给模数寄存器一个有效的匹配。

TPM 通道寄存器中的输出比较值决定 CPWM 信号 （图 15-16）的脉宽 （占空比）。如果 ELSnA = 0，当计数增大强制 CPWM 输出信号为低时进行一个比较，当计数减小强制输出为低时进行一个比较。计数器计数增加直到它达到 TPMxMODH:TPMxMODL 中的模数设定值，然后计数下降直至为零。这将设置周期等于两倍 TPMxMODH:TPMxMODL。

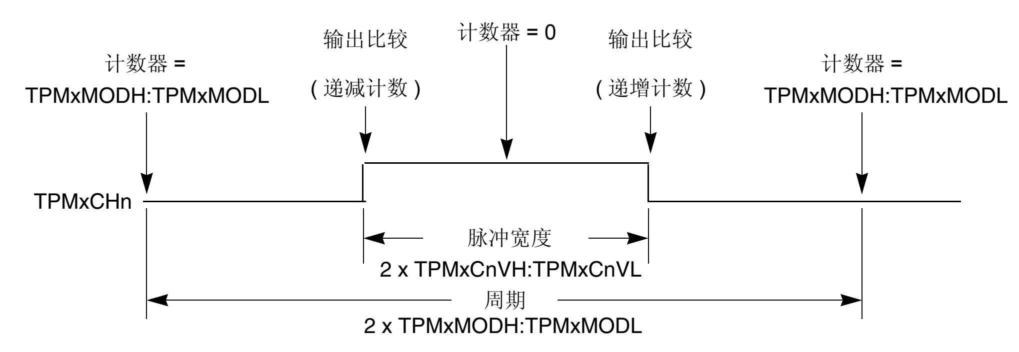


图 15-16. 16CPWM 周期和脉冲宽度 （ELSnA=0）

中心对齐 PWM 输出比边沿对齐 PWM 模式产生更少的噪音，这是因为较少的 I/O 引脚转换被列队在同一个系统时钟边沿。某些需要电机驱动的类型也需要这种 PWM 类型。

当计数器操作在增 / 减计数模式时，输入捕捉、输出比较和边沿对齐 PWM 功能无意义，这表明当 CPWMS = 1 时，所有带 TPM 的活跃通道必须运行于 CPWM 模式。

TPM 可被用在 8 位 MCU。定时器通道寄存器中的设置被缓存以确保 16 位一致更新和避免产生不可预料的 PWM 脉宽。写 TPMxMODH、 TPMxMODL、 TPMxCnVH 和 TPMxCnVL 实际上就是写缓冲寄存器。

在中心对齐 PWM 模式下，根据 CLKSB: CLKSA 位的值用写缓冲里的值来更新 TPMxCnVH:L 寄存器。所以：

- 如果 （CLKSB: CLKSA=0:0），当第二个字节被写时寄存器被更新
- 如果（CLKSB: CLKSA!=0:0），在两个字节都被写后， TPM 计数器从（TPMxMODH : TPMxMODL - 1 ）变为 （TPMxMODH : TPMxMODL）时寄存器被更新。如果 TPM 计数器是一个自运行计数器则当 TPM 计数器从 0XFFFE 变为 0XFFFF 时更新被执行。

写 TPMxSC 会取消任何写入 TPMxMODH 和 / 或 TPMxMODL 的值，并且复位模寄存器一致机制。写 TPMxCnSC 会取消写入通道值寄存器的值，并且复位 TPMxCnVH : TPMxCnVL 的一致机制。

15.7 复位概述

15.7.1 概要

MCU 复位会导致 TPM 复位。

15.7.2 复位操作

复位清零 TPMxSC 寄存器并禁止 TPM 时钟和定时器溢出中断 （TOIE=0）。 CPWMS、 MSnB、MSnA、ELSnB、 ELSnA 被清零，为不带 I/O 引脚逻辑的相关引脚的输入捕捉操作配置所有 TPM 通道 （所以所有与 TPM 相关的 MCU 引脚恢复为通用 I/O 引脚）。

15.8 中断

15.8.1 概述

TPM 给核心计数器提供一个溢出中断，并给每个通道提供一个中断。该通道中断意义取决于每个通道的操作模式。如果通道被配置为输入捕捉，则每次设定的输入捕捉边沿被确认时中断标志被置位。如果通道被配置为输出比较或 PWM 模式，当核心定时器计数器等于 16 位通道值寄存器值，中断被置位。

所有 TPM 中断被列在表 15-9，包括中断名称，中断使能。这些中断使能可阻塞离开 TPM 的中断请求和通过独立中断处理逻辑获得确认。

表 15-9. 中断

中断名称	中断使能名称	触发源	描述
TOF	TOIE	计数器溢出	每次定时器计数器达到它的最终计数值时置位 （可转向下个计数值通常为 0x0000）
CHnF	ChnIE	通道事件	一个输入捕捉或输出比较事件发生在通道 n

TPM 模块将提供一个高保真中断信号。中断向量和优先级由终端模块芯片集成时间决定，可参考用户手册的中断模块章节或芯片完整文档了解相关的细节。

15.8.2 中断操作描述

对于 TPM 的每个中断源，当中断情形，如定时器溢出，通道输入捕捉，或输出比较事件被确定，将对一个标志位置位。该标志位可通过软件读取，以确定事件已发生，或者设置相关使能为允许硬件中断。当中断使能位被置位，只要相关中断标志等于 1，将产生一个相应的中断。用户软件必须执行一些步骤序列，用以在终端服务例程返回之前清空中断标志。

清空 TPM 中断标志的流程包括两个步骤，即当标志位为 1 时读出它然后将它置 0。如果在这两步处理之间检测到新事件，该序列被复位，在第二步后中断标志位保持置位以避免丢失新事件。

15.8.2.1 定时器溢出中断 （TOF）

TOF 中断操作的细节和方式取决于 TPM 系统操作的模式 （通用定时功能对应于中心对齐 PWM）。该标志位可通过上述两步处理来清零。

15.8.2.1.1 通常情形

通常情况下，当定时器计数器从 0xFFFF 变为 0x0000 时 TOF 被置位。当 TPM 未被配置为中心对齐 PWM 模式 （CPWMS=0）时，在定时器计数值从终点值 （模数寄存器中的值）变为 0x0000 时 TOF 置位。这种情形为计数器溢出一般方式。

15.8.2.1.2 中心对齐 PWM 情形

CPWMS = 1 时，当定时器计数器在终点值（模数寄存器中的值）处改变计数方向从递增转为递减时，TOF 置位。这种情形下， TOF 对应一个 PWM 周期的末尾。

15.8.2.2 通道事件中断描述

通道中断方式取决于通道当前模式（输入捕捉，输出比较，边对齐 PWM。中心对齐 PWM）。

15.8.2.2.1 输入捕捉事件

当通道被配置为输入捕捉通道时， ELSnB:ELSnA 控制位选择无边沿，上升沿，下降沿或任意边沿作为触发输入捕捉事件边沿。当检测到被选择边沿时，中断标志位被置位。该标志位可通过在 15.8.2 节 中断操作描述中的两步流程来清零。

15.8.2.2.2 输出比较事件

若通道被配置为输出比较通道，当核心定时器计数器值等于通道值寄存器 16 位值时，中断标志位被置位。该标志位可通过 15.8.2 节 中断操作描述中的两步流程来清零。

15.8.2.2.3 PWM 结束占空比事件

若通道被配置为 PWM，则可能的操作有两种。当通道被配置为边沿对齐 PWM 时，在定时器计数器等于通道值寄存器 (标记活跃占空比周期末尾) 时通道标志被置位。当通道被配置为中心对齐 PWM 时，在每个 PWM 周期内，定时器计数器两次等于通道值寄存器的值。在这种 CPWM 情形下，在占空比周期的开始和结尾时通道标志被置位，此时，定时器计数器的值等于通道寄存器的值。该标志位可通过在 15.8.2 节 中断操作描述中的两步流程来清零。

第 16 章
开发支持

16.1 介绍

HCS08 中的开发支持系统包括后台调试控制器 (BDC) 和片上调试模块 (DBG)。BDC 提供单线调试接口，与目标 MCU 连接，通过这个接口可以方便地进行片上闪存和其它非易失性存储器的编程。 BDC 也是开发用的主要调试接口，允许以非侵入式方式访问存储器数据和传统调试功能，如 CPU 寄存器修改、断点和单指令跟踪命令等。

在 HCS08 产品系列中，在外部管脚得不到地址和数据总线信号 （即使在测试模式也不包括）。通过单线后台调式接口向目标 MCU 传输命令来实现调试。调试模块提供了一种有选择性地触发和捕获总线信息的方式，这样外部开发系统可以对 MCU 内发生的事件按周期进行重建，而不需要从外部访问 MCU 的地址和数据信号。

ICGLCLK 是 MC9S08AC60 系列备选的 BDC 时钟源。有关 ICGCLK 及如何选择时钟源的更多信息，请参见 第 10 章 内部时钟发生器 (S08ICGV4)。

16.1.1 特性

BDC 模块具有以下特性：

- 单引脚进行模式选择和后台调试通信
- BDC 的寄存器不位于存储器地址中
- SYNC 命令确定目标通信速率
- 非侵入式命令访问存储器
- 激活后台模式命令可访问 CPU 寄存器
- GO 和 TRACE1 命令
- BACKGROUND 命令可以将 CPU 从停止模式或等待模式中唤醒
- BDC 内置一个硬件地址断点
- 如果 BDC 使能，则振荡器运行在停止模式
- 处于激活后台模式时 , COP 看门狗禁止

ICE 系统具有以下特性：

- 两个触发比较器：两个地址 + 读 / 写 (R/W) 或一个完整地址 + 数据 + R/W
- 灵活的 8 字 x 16 位 FIFO (先进先出) 缓存 , 用于捕获信息：
  - 流程变化的地址 或
  - 纯事件数据
- 两类断点：
  - 指令操作码的标记断点
  - 访问任何地址的强制断点
- 九种触发模式：
  - 基本：只有 A，A 或 B
  - 顺序：A 然后 B
  - 全部：A 并且 B 数据，A 并且非 B 数据
  - 事件 (存储数据): 纯事件 B， A 然后纯事件 B
  - 范围：在范围以内 (A ≤ 地址≤ B), 在范围以外 (地址 < A 或地址 > B)

16.2 后台调试控制器 (BDC)

HCS08 系列中的所有 MCU 都包含一个单线后台调试接口，此接口支持片上非易失性存储器的在线编程和先进的非侵入式调试功能。与早期的 8 位 MCU 的调试接口不同 , 这个系统不干扰正常的应用资源。它不使用任何用户存储器或存储器映射中的地址，也不分享任何片上外设。

BDC 命令分为两大组：

- 激活后台模式命令要求目标 MCU 处于激活后台模式 (用户程序未运行)。激活后台调试模式命令允许读写 CPU 寄存器，允许用户一次跟踪一个用户指令，或从激活后台模式进入用户程序。
- 非侵入式命令可以随时执行，即使用户的程序正在运行。
  非侵入式命令允许用户在后台调试控制器中读写 MCU 存储器位置或访问状态和控制寄存器。

通常，可以用相对简单的接口盒将来自主机的命令转换为用于连接单线后台调试系统的串口所需的命令。根据不同的开发工具供应商，这个接口盒可能采用标准 232 串行端口，或是并行打印端口，或是其它类型的

通信端口（例如 USB 接口），来与 PC 主机通信。这个接口盒一般通过接地、BKGD 引脚、RESET，有时还有 V_{DD} 与目标系统连接。RESET 引脚的开漏连接允许主机强制目标系统复位，这有助于重新获得对目标系统的控制，或在片上非易失性存储器重新编程之前，控制目标系统的启动。有时可以用 V_{DD} 来允许接口盒使用目标系统的电源，避免再需要一个单独的电源。但是，如果单独对接口盒供电，它可以连接到一个正在运行的目标系统，而不必强制目标系统复位，否则会干扰正在运行的应用程序。

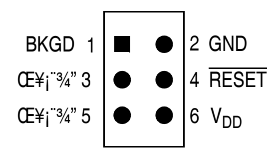


图 16-1. BDM 工具连接器

16.2.1 BKGD 引脚描述

BKGD 是单线后台调试接口引脚。这个引脚的主要功能是实现激活后台模式命令和数据的双向串行通信。在复位过程中，这个引脚用来选择启动激活后台模式或启用用户的应用程序。这个引脚还用来请求定时同步响应脉冲，允许主机开发工具确定后台调试串行通信的正确时钟频率。

BDC 串行通信采用首先引入 M68HC12 系列微处理器上的定制串行协议。这个协议假定主机知道由目标 BDC 时钟率决定的通信时钟速率。所有通信通过主机启动和控制，主机驱动高到低边沿发出每个位时间开始信号。命令和数据以最重要的位先发 (MSB 先发) 的方式发送。有关通信协议的详细信息，请参见 16.2.2 节 通信详细介绍。

如果主机偿试与 BDC 时钟速率未知的目标 MCU 通信，可以发送 SYNC 命令给目标 MCU ，请求定时同步响应信号，通过这个信号，主机可以判断正确的通信速率。

BKGD 是伪开漏引脚，有一个片上上拉，因此不需要外部上拉电阻。与典型的开漏引脚不同，引脚上的外部 RC 时间常数 （受外部容性的影响），在信号上升时间上几乎不起作用。定制协议提供瞬态加速脉冲，强制提高这个引脚的上升时间，而没有驱动电平冲突风险。参见 16.2.2 节 通信详细介绍，了解更多详情。

当 6 引脚的 BDM 接口连接器没有连接调试盒时，BKGD 的内部上拉会选择正常的操作模式。当调试盒连接到 BKGD 时，可以在 MCU 复位后强制它进入激活后台调试模式。强制激活后台调试的具体条件取决于 HCS08 衍生产品 (参见 “开发支持”小节的介绍)。不必复位目标 MCU 来通过后台调试接口来与之通信。

16.2.2 通信详细介绍

BDC 串行接口需要外部控制器来生成 BKGD 引脚上的下降沿，指示每个位时间的开始。无论数据是发送或接收，外部控制器都会提供这个下降边沿。

BKGD 是伪开漏引脚，它可以被外部控制器或 MCU 驱动。数据以 MSB 先发的形式且以每位 16 个 BDC 时钟周期的速率 (标定速率) 发送。如果来自主机的下降边沿之间产生 512 BDC 时钟周期，则该接口超时。出果出现超时，任何正在进行的 BDC 命令被中止，对目标 MCU 系统的存储器或操作模式没有影响。

定制串行协议要求调试盒知道目标 BDC 通信时钟速率。

BDC 状态和控制寄存器中的时钟开关 (CLKSW) 控制位允许用户选择 BDC 时钟源。BDC 时钟源可以是总线，或备用的 BDC 时钟源。

BKGD 引脚可以接收高或低电平，或发送高或低电平。下图显示了每种情况的时序。接口时序与目标 BDC 中的时钟同步，但是与外部主机异步。显示的内部 BDC 时钟信号是计数周期的参考。

在图 16-2 中，外部主机将逻辑 1 或 0 发送到目标 HCS08 MCU 的 BKGD 引脚。主机与目标异步，因此主机生成的 BKGD 下降边沿与目标所认为的位时间起始点有 0- 到 -1 周期的延迟。 10 个目标 BDC 时钟周期后，目标获得 BKGD 引脚的电平。通常，主机在主机到目标方向的传输过程中驱动伪开漏 BKGD 引脚 ，以加快上升边沿。由于目标在主机至目标方向的传输周期中不驱动 BKGD 引脚，因此没有必要在此期间将线路作为开漏信号。

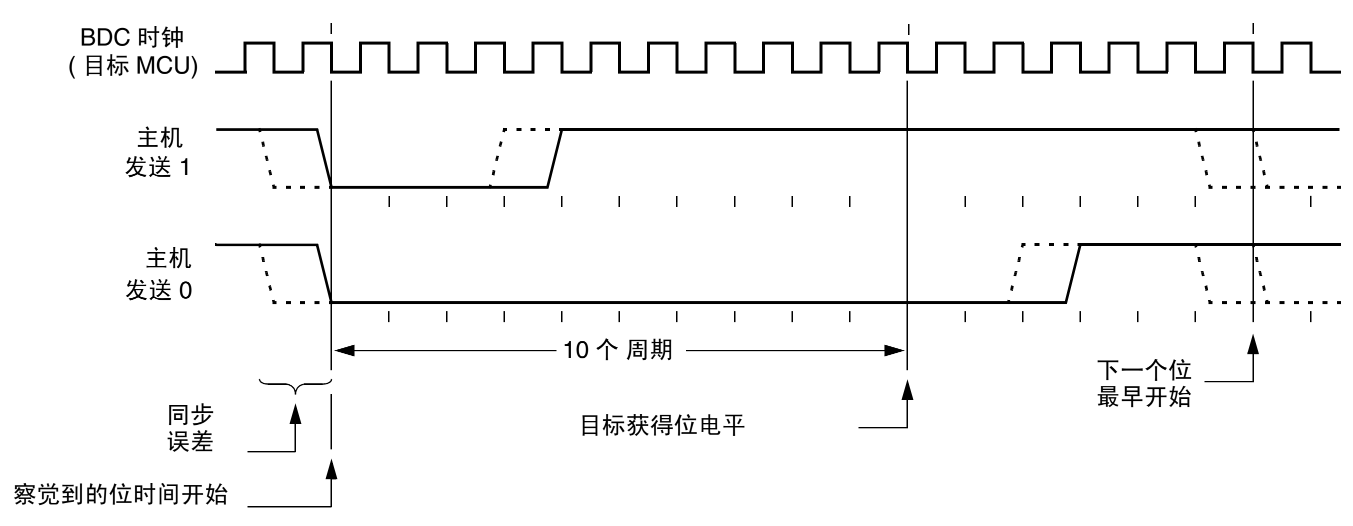


图 16-2. BDC 主机到目标方向串行位时序

图 16-3 显示主机从目标 HCS08 MCU 收到逻辑 1。由于主机与目标异步，因此主机生成的 BKGD 上的下降边沿与目标 MCU 所认为的位时间起始点有 0 到 1 个周期的延迟。主机保持低 BKGD 引脚足够长的时间，使目标识别它 （至少两个目标 BDC 周期）。目标 MCU 在其认为的位计时开始七个周期后驱动瞬时高态加速脉冲，在此之前，主机必须释放低电平驱动。主机必须在其启动位时间约 10 个周期后采样位电平。

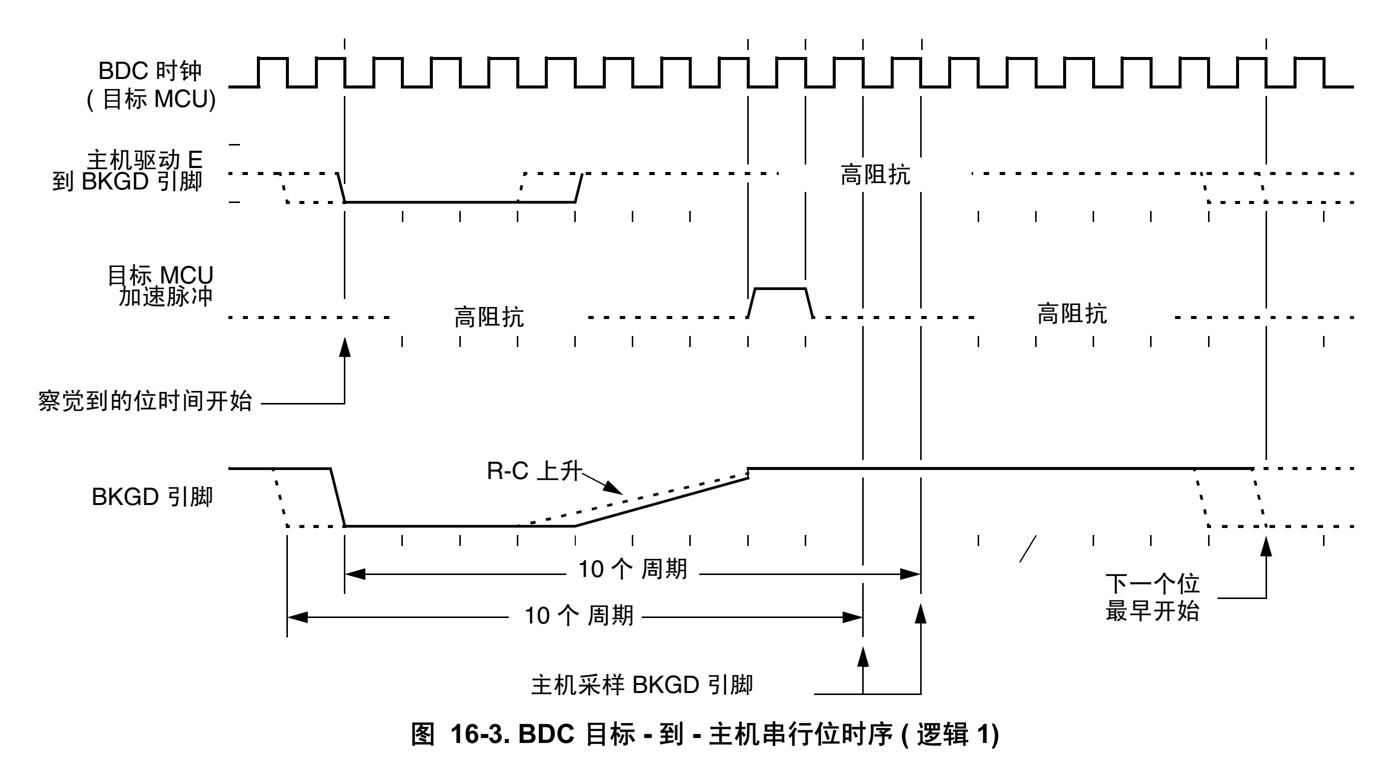


图 16-3. BDC 目标 - 到 - 主机串行位时序 (逻辑 1)

在图 16-4 中，主机从目标 HCS08 MCU 收到逻辑 0。由于主机与目标异步，因此主机生成的 BKGD 上的下降边沿与目标 MCU 所认为的位时间起始点有 0 到 1 个周期的延迟。主机启动位时间，但是目标 HCS08 MCU 完成它。由于目标 MCU 希望主机接收逻辑 0，它保持低 BKGD 引脚 13 个 BKGD 时钟周期，然后驱动引脚置高，加速上升沿。主机在启动位时间约 10 个周期后采样位电平。

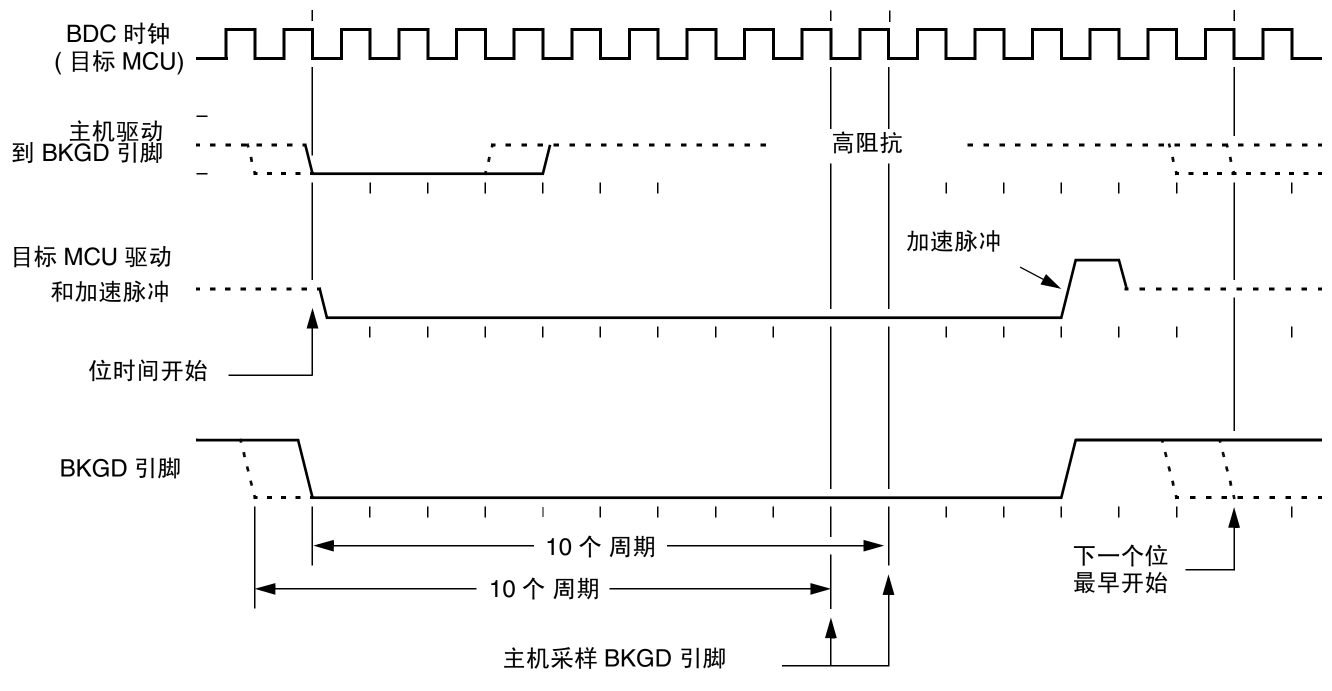


图 16-4. BDM 目标 - 到 - 主机串行位时序 (逻辑 0)

16.2.3 BDC 命令

BDC 命令以串行形式从主机发送到目标 HCS08 MCU 的 BKGD 引脚。所有命令和数据都采用定制 BDC 通信协议以 MSB 先发的形式发送。激活后台调试模式命令要求目标 MCU 当前处于激活后台模式，而非侵入式命令可以随时发出，无论目标 MCU 是处于激活后台模式还是运行用户应用程序。

表 16-1 列出了所有 HCS08 BDC 命令，并简要描述了它们的编码结构，以及每个命令的含义。

16.2.3.1 编码结构术语

在表 16-1 中使用了以下术语来描述 BDC 命令的编码结构。

命令在主机 - 目标 MCU 方向上从一个 8 位十六进制命令代码 （MSB 先发）开始

/	=	将命令的各部分分开
d	=	延迟 16 个目标 BDC 时钟周期
AAAA	=	主机 - 目标 MCU 方向上的一个 16 位地址
RD	=	目标 MCU- 主机方向上的 8 位读数据
WD	=	主机 - 目标 MCU 方向上的 8 位写数据
RD16	=	目标 MCU- 主机方向上的 16 位读数据
WD16	=	主机 - 目标 MCU 方向上的 16 位写数据
SS	=	目标 MCU- 主机方向 (STATUS) 上的 BDCSCR 内容
CC	=	主机 - 目标 MCU 方向 (CONTROL) 方向上的 8 位写数据
RBKP	=	目标 MCU - 主机方向 (从 BDCBKPT 断点寄存器) 上的 16 位读数据
WBKP	=	主机 - 目标 MCU 方向 (至 BDCBKPT 断点寄存器)16 位写数据

表 16-1. BDC 命令

命令术语	激活 BDM/ 非侵入式	编码结构	描述
SYNC	非侵入式	n/a[1]	请求定时参考脉冲来确定目标 BDC 通信速率
ACK_ENABLE	非侵入式	D5/d	使能应答协议。参见飞思卡尔文档 HCS08RMv1/D。
ACK_DISABLE	非侵入式	D6/d	禁止响应协议。参见飞思卡尔文档 HCS08RMv1/D。
BACKGROUND	非侵入式	90/d	如果使能，则进入激活后台调试模式 (如果 ENBDM 位等于 0，则忽略)
READ_STATUS	非侵入式	E4/SS	从 BDCSCR 读取 BDC 状态
WRITE_CONTROL	非侵入式	C4/CC	向 BDCSCR 中写入对 BDC 的控制
READ_BYTE	非侵入式	E0/AAAA/d/RD	从目标存储器读取字节
READ_BYTE_WS	非侵入式	E1/AAAA/d/SS/RD	读字节和报告状态
READ_LAST	非侵入式	E8/SS/RD	从地址重新读字节，仅读和报告状态
WRITE_BYTE	非侵入式	C0/AAAA/WD/d	将一个字节写入到目标存储器
WRITE_BYTE_WS	非侵入式	C1/AAAA/WD/d/SS	写入一个字节并报告状态
READ_BKPT	非侵入式	E2/RBKP	读 BDCBKPT 断点寄存器
WRITE_BKPT	非侵入式	C2/WBKP	写 BDCBKPT 断点寄存器
GO	激活 BDM	08/d	从 PC 当前的地址执行用户应用程序
TRACE1	激活 BDM	10/d	在 PC 的地址跟踪 1 条用户指令，然后返回到激活后台模式
TAGGO	激活 BDM	18/d	与 GO 相同，但激活外部标签 (HCS08 器件没有外部标签引脚)
READ_A	激活 BDM	68/d/RD	读累积器 (A)
READ_CCR	激活 BDM	69/d/RD	读条件代码寄存器 (CCR)
READ_PC	激活 BDM	6B/d/RD16	读程序计数器 (PC)
READ_HX	激活 BDM	6C/d/RD16	读 H 和 X 寄存器对 (H:X)
READ_SP	激活 BDM	6F/d/RD16	读堆栈指针 (SP)
READ_NEXT	激活 BDM	70/d/RD	以 1 为单位递增 H:X ， 然后读位于 H:X 的存储器字节
READ_NEXT_WS	激活 BDM	71/d/SS/RD	以 1 为单位递增 H:X，然后读位于 H:X. 的存储器字节。并报告状态和数据。
WRITE_A	激活 BDM	48/WD/d	写累积器 (A)
WRITE_CCR	激活 BDM	49/WD/d	写条件代码寄存器 (CCR)
WRITE_PC	激活 BDM	4B/WD16/d	写程序计数器 (PC)
WRITE_HX	激活 BDM	4C/WD16/d	写 H 和 X 寄存器对 (H:X)
WRITE_SP	激活 BDM	4F/WD16/d	写堆栈指针 (SP)
WRITE_NEXT	激活 BDM	50/WD/d	以 1 为单位递增 H:X ， 然后写位于 H:X 的存储器字节。
WRITE_NEXT_WS	激活 BDM	51/WD/d/SS	以 1 为单位递增 H:X ， 然后写位于 H:X 的存储器字节。报告状态。

[1] SYNC 命令是特殊操作，不需要命令代码。

SYNC 命令与其它 BDC 命令不同，因为主机在分析完对 SYNC 命令的响应之前，不需要知道 BDC 通信的速率。

要发出 SYNC 命令，主机：

- 保持 BKGD 引脚为低电平至少 128 周期，而且是以最慢的 BDC 时钟来计 (最慢的时钟一般是参考振荡器 /64 或自时钟速率 /64。)
- 驱动 BKGD 达到高电平，实现瞬态加速，快速上升时间 (这个加速脉冲一般是系统中最快的时钟的一个周期)
- 去除 BKGD 引脚的所有驱动，这样它可回复到高阻抗。
- 监视 BKGD 引脚得到同步响应脉冲

当检测到主机的 SYNC 请求 （比在正常 BDC 通信过程中发生的慢时钟要长），则目标 MCU：

- 等待 BKGD 返回到逻辑高电平
- 延迟 16 个周期，以供主机停止驱动高电平加速脉冲
- 驱动 BKGD 低态 128 BDC 时钟周期
- 驱动一个周期的高电平加速脉冲，在 BKGD 上强制快速上升时间
- 去除 BKGD 引脚的所有驱动，这样它可回复到高阻抗。

主机测量这个 128 周期的响应脉冲的低电平时间，确定速率，用于后续的 BDC 通信。主机一般可以确定的通信速率与实际目标速率的误差只有百分之几，这是通信协议能够接受的。

16.2.4 BDC 硬件断点

BDC 包括一个相对简单的硬件断点，将 CPU 地址总线与 BDCBKPT 寄存器中的 16 位匹配值进行比较。这个断点可以生成强制断点或标记断点。强制断点使 CPU 在存取断点地址后的第一个指令边界进入激活后台调试模式。标记的断点使指令操作码在断点地址被标记，这样当 CPU 到达指令队列的终点时，将进入激活后台模式，而不是执行该指令。这意味着标记的断点可能放置在指令操作代码的地址上，而强制断点可以设置在任何地址。

BDC 状态和控制寄存器 (BDCSCR) 中的断点使能 (BKPTEN) 控制位用来激活断点逻辑 (BKPTEN = 1)。当 BKPTEN = 0 （复位后它的默认值），断点逻辑禁止，无论其它 BDC 断点中的值是多少，也不管控制位如何，均不请求断点。 BDCSCR 中的强制 / 标记选择 （FTS）控制位用来选择强制 (FTS = 1) 或标记 (FTS = 0) 类型断点。

片上调试模块 (DBG) 包括两个额外的硬件断点的电路，这两个硬件断点比 BDC 模块中的简单断点更灵活。

16.3 片上调试系统 (DBG)

由于 HCS08 器件没有外部地址和数据总线，在线仿真器最重要的功能已经构建在 MCU 的芯片上。这种调试系统包含可以灵活地存储地址或数据总线信息的 8 级 FIFO，和一个确定何时捕获总线信息以及捕获哪些总线信息的灵活触发系统。这个系统依赖单线后台调试系统来存取调试控制寄存器，读取 8 级阶 FIFO 的结果。

调试模块包括控制和状态寄存器，可以在用户存储器映射中存取。这些寄存器位于高地址寄存器空间中，避免使用宝贵的直接页面存储器空间。

大多数调试模块的功能在开发过程使用，用户程序很少存取调试模块的任何控制和状态寄存器。一个例外就是调试系统可以提供一种手段来实施某种形式的 ROM 补丁。 16.3.6 节 硬件断点中对此有更详细的描述。

16.3.1 比较器 A 和 B

两个 16 位比较器器 (A 和 B) 可以选择用 R/W 信号或一个操作码跟踪电路来鉴定。比较器单独的控制位允许你忽略每个比较器的 R/W。操作码跟踪电路可选地允许你规定，如果操作码在规定的地址实际执行，而不是只从存储器读到指令队列中，则触发将发生。比较器还能够进行庞大的比较，支持范围内和范围外触发模式。在所有 BDC 存取过程中，比较器临时禁止。

比较器 A 总是与 16 位 CPU 地址相关联。比较器 B 根据所选的触发模式比较 CPU 地址或 8 位 CPU 数据总线。由于 CPU 数据总线分为只读数据和写数据总线， RWAEN 和 RWA 控制位有一个额外的目的，在完整地址加上数据比较中，它们被用来确定其中哪些总线用在比较器 B 数据总线比较中。如 RWAEN = 1 (激活)，RWA = 0 (写), 则使用 CPU 的写数据总线，否则用 CPU 的只读数据。

当前选择触发模式确定当比较器检测到合格的匹配条件时，调试器逻辑做什么。匹配可以导致以下情况：

- 生成 CPU 断点
- 将数据总线值存储到 FIFO 中
- 开始将流变化地址存储到 FIFO 中 (开始类型跟踪)
- 停止将流变化地址存储到 FIFO 中 （结束类型跟踪）

16.3.2 总线捕获信息和 FIFO 操作

使用 FIFO 的通常方式是建立触发模式和其它控制选项，然后打开调试器。当 FIFO 填满后，或调试器停止将数据存储到 FIFO 后，你可以按信息存储的顺序从中读取信息。状态位指示数据所在的 FIFO 中的有效信息的字数。如果在 满 （CNT = 1:0:0:0）之前将 ARM 写为 0，以人工停止跟踪，信息移动一个位置，主机必须执行 ((8 - CNT) - 1)FIFO 虚读操作，使信息进入到 FIFO 中的第一个重要入口。

在大多数触发模式中，存储在 FIFO 中的信息包含 16 位流变化地址。在这些情况中，先读 DBGFH 然后读 DBGFL，从 FIFO 中获得一个一致的信息字。读 DBGFL (FIFO 数据端口的低阶字节) 会使 FIFO 移动，这样下一个信息字可以在 FIFO 数据端口提供。在纯事件触发模式（参见 16.3.5 节 触发模式）中， 8 位数据信息存储在 FIFO 中。在这些情况中， FIFO(DBGFH) 的上半部分没有被使用，仅仅通过读 DBGFL 来从 FIFO 中读出数据。每次读 DBGFL 时， FIFO 都会移动，这样通过 DBGFL 的 FIFO 数据端口可以获得下一个数据值。

在触发模式中， FIFO 保存流变化地址， CPU 地址与 FIFO 的输入端有一个延迟。由于这个延迟，如果触发事件本身是一个流变化地址或在触发事件启动 FIFO 后下两个周期中出现了流变化地址，它将不保存在 FIFO 中。如果是结束 - 跟踪的情况，当触发事件是一个流变化，则它将保存为运行的调试器的最后一个流变化入口。

当调试器没有打开时，FIFO 还可以用来生成所执行指令地址的分析。当 ARM = 0, 读 DBGFL 会使最近获取的操作码的地址保存在 FIFO 中。采用分析功能，主机调试器将从 FIFO 中读取地址，即以常规的间隔先读 DBGFH 然后读 DBGFL，。前 8 个值将被丢弃，因为它们对应于初始需要填充 FIFO 的 8 个 DBGFL 读取。DBGFH 和 DBGFL 的其它周期读取则返回关于所执行指令的延迟信息，这样主机调试器可以对执行指令地址进行分析。

16.3.3 流变化信息

为了减少存储在 FIFO 中的信息数量，只保存与使正常的指令执行顺序发生变化的指令相关的信息。知道存储在目标系统中的源和对象代码程序后，外部调试器可以通过来自 FIFO 中存储的大量流变化信息的许多指令来重现执行路径。

对于采用了分支的条件分支指令 （分支条件为真），则保存源地址 (条件分支操作码的地址)。由于 BRA 和 BRN 指令不是条件的，这些事件不会使流变化信息存储在 FIFO 中。

间接 JMP 和 JSR 指令采用 H:X 间址寄存器对的当前内容，确定目的地址，这样调试系统为任何间接 JMP 或 JSR 保存运行时的目的地址。对于中断， RTI 或 RTS, 目的地址作为流变化信息存储在 FIFO 中。

16.3.4 标记 vs. 强制断点和触发器

标记一词指当指令操作码被取到指令队列时识别它，但是不采取任何其它操作，直到且除非指令被 CPU 真正执行。这种区分非常重要，因为任何因跳转、分支、子例程调用、或中断而发生的流变化都会导致一些指令被取到指令队列，未执行就被丢弃。

强制类型的断点等待当前指令完成，然后执行断点请求操作。通常操作是进入激活后台调试模式，而不是继续用户应用程序中的下一个指令。

标记 vs. 强制这一术语在调试模块的两种情况下使用。第一种情况指从调试模块向 CPU 发送断点请求。第二种情况指从比较器向调试控制逻辑发送匹配信号。当标记类断点发送给 CPU 时，信号与操作码一起进入指令队列，这样当这个操作码被执行时， CPU 将有效地用 BGND 操作码代替被标记的操作码，这样 CPU 进入激活后台调试模式，而不是执行被标记的指令。当 DBGT 寄存器中的 TRGSEL 控制位被设置为选择标记类操作，比较器 A 或 B 的输出被调试模块中的逻辑块鉴定，这个逻辑块跟踪操作码，如果比较地址的操作码被实际执行，则只向该调试器生成一个触发。每个比较器都有单独的操作码跟踪逻辑，这样整个指令队列一次不只一个比较事件被跟踪。

16.3.5 触发模式

触发模式控制调试器运行的整体行为。DBGT 寄存器中的 4 位 TRG 字段选择九个触发模块中的一个。当 DBGT 寄存器中的 TRGSEL = 1, 比较器的输出必须在触发 FIFO 操作前通过操作码跟踪电路传播。DBGT 中的 BEGIN 位选择当检测到合格的触发时 FIFO 是否开始存储数据 （开始跟踪），或 FIFO 从其打开之时开始循环存储数据，直到检测到合格的触发 （结束触发）。

将 1 写入到寄存器中的 ARM 位便可启动调试运行，它设置 DBGS 中的 ARMF 标记，并清除 AF 和 BF 标记及 CNT 位。开始跟踪调试运行当 FIFO 满时结束。结束跟踪运行则在所选触发事件发生时结束。任何调试运行均可通过将 0 写入到 DBGC 中的 ARM 或 DBGEN 位停止。

除纯事件模式外的所有触发模式中，FIFO 都存储流变化地址。在纯事件触发模式中，FIFO 将数据存储在 FIFO 的八低八位。

控制位在纯事件触发模式中被忽略，而且所有这样的调试运行都是开始类型跟踪。当 TRGSEL = 1 选择操作码获取触发器，没有必要在比较中使用 R/W，因为操作码标签只应用于操作码获取，而这一直都是读周期。在采用全模式触发器时，规定 TRGSEL = 1 也是不正常的，因为操作码的值通常在特定的地址可以知道。

下面的触发模式描述只说明了导致触发的主要比较器条件。比较器 A 或 B 通常都可以被 R/W 进一步鉴定，通过将 RWAEN (RWBEN) 和相应的 RWA (RWB) 值设置为与 R/W 相匹配。如果 BRKEN = 1，来自比较器的带可选 R/W 鉴定的信号，用来请求 CPU 断点， TAG 决定 CPU 请求是标记请求还是强制请求。

只 A— 当地址匹配比较器 A 的值时触发

A 或 B — 当地址匹配比较器 A 或 B 的值时触发

A 然后 B — 当地址匹配比较器 B 但只能在另一个周期的地址匹配比较器 A 的值以后，触发。可能在 A 匹配后 B 匹配前有许多周期。

A 和 B 数据 （全模式）— 这称为全模式，因为地址，数据和 R/W (可选) 必须在同一个总线周期内匹配，才能产生触发事件。比较器 A 检查地址，比较器的低阶字节检查数据，如果 RWAEN = 1， R/W 对照 RWA 进行检查。比较器 B 的高半部分没有使用。

在全触发模式中，规定标签类 CPU 断点 (BRKEN = TAG = 1) 没有用，但是如果你这样做了，就会忽略比较器 B 数据匹配，以例向 CPU 发送标签请求，当比较器 A 地址匹配时发送 CPU 断点。

A 但非 B 数据 （全模式）— 地址必须匹配比较器 A, 数据必须不能匹配比较器 B 的低阶部分，如果 RWAEN = 1， R/W 必须匹配 RWA。所有三个条件必须在同一个总线周期中达到才能引起触发。

在全触发模式中，规定标签类 CPU 断点 (BRKEN = TAG = 1) 没有用，但是如果你这样做了，就会忽略比较器 B 数据匹配，以例向 CPU 发送标签请求，当比较器 A 地址匹配时发送 CPU 断点。

纯事件 B （存储数据）— 当地址每次匹配比较器 B 的值时，触发事件发生。触发事件导致数据被捕获到 FIFO 中。当 FIFO 满时调试运行结束。

A 然后纯事件 B （存储数据）— 当地址匹配比较器 A 中的值后，每次地址匹配比较器 B 中的值时，触发事件发生。触发事件导致数据被捕获到 FIFO 中。当 FIFO 满时调试运行结束。

范围内 （A ≤ 地址≤ B）— 当地址大于或等于比较器 A 的值，且小于等于比较器 B 的值时，触发发生。

范围外 （地址 < A 或 地址 > B） — 当地址小于比较器 A 的值，或大于比较器 B 的值时，触发发生。

16.3.6 硬件断点

DBGC 寄存器中的 BRKEN 控制位可以设置为 1，来允许使用 16.3.5 节 触发模式所描述的任何触发条件 , 向 CPU 生成硬件断点请求。 DBGC 中的 TAG 控制断点请求是否处理为标记类断点或强制类断点。标记断点使当前的操作码进入指令队列时被标记。如果标记的操作码达到队列的末端， CPU 执行 BGND 指令，进入激活后台调试模式，而不是执行被标记的操作码。强制类断点使 CPU 完成当前指令，然后进入激活后台调试模式。

如果后台调试模式没有被通过 BKGD 引脚的串行 WRITE_CONTROL 命令激活 (ENBDM = 1), CPU 将执行 SWI 指令，而不是进入激活后台调试模式。

16.4 寄存器定义

本小节描述了 BDC 和 DBG 寄存器及控制位。

参见本文的器件概述章节中的 高位页面寄存器一览，了解所有 DBG 寄存器的绝对地址分配。本小节只按名字参考了寄存器和控制位。使用飞思卡尔提供的等式或头文件，将这些名称翻译为相应的绝对地址。

16.4.1 BDC 寄存器和控制位

BDC 有两个寄存器：

- 状态和控制寄存器 (BDCSCR) 是一个包含后台调试控制器控制和状态位的 8 位寄存器。
- BDC 断点匹配寄存器 (BDCBKPT) 拥有一个 16 位断点匹配地址。

这些寄存器通过专门的串行 BDC 命令接入，没有位于目标 MCU 的存储器空间中 (因此，它们没有地址，用户程序不能接入)。

BDCSCR 中的一些位有写限制，否则这些寄存器可以随时被读或写。例如，当 MCU 处于激活后台调试模式中时，ENBDM 控制位不能被写。 (这防止了在 MCU 已经处于激活后台模式时，禁止激活后台模式的控制位的模糊条件) 而且，有四个状态位 (BDMACT, WS, WSF, 和 DVF) 是只读状态指示符，永远也不能被 WRITE_CONTROL 串行 BDC 命令写。时钟开关 (CLKSW) 控制位随时都可读或写。

16.4.1.1 BDC 状态和控制寄存器 (BDCSCR)

这个寄存器可以被串行 BDC 命令 (READ_STATUS 和 WRITE_CONTROL) 读或写，但是用户程序不能存取它，因为它不位于 MCU 的正常的存储器映射空间中。

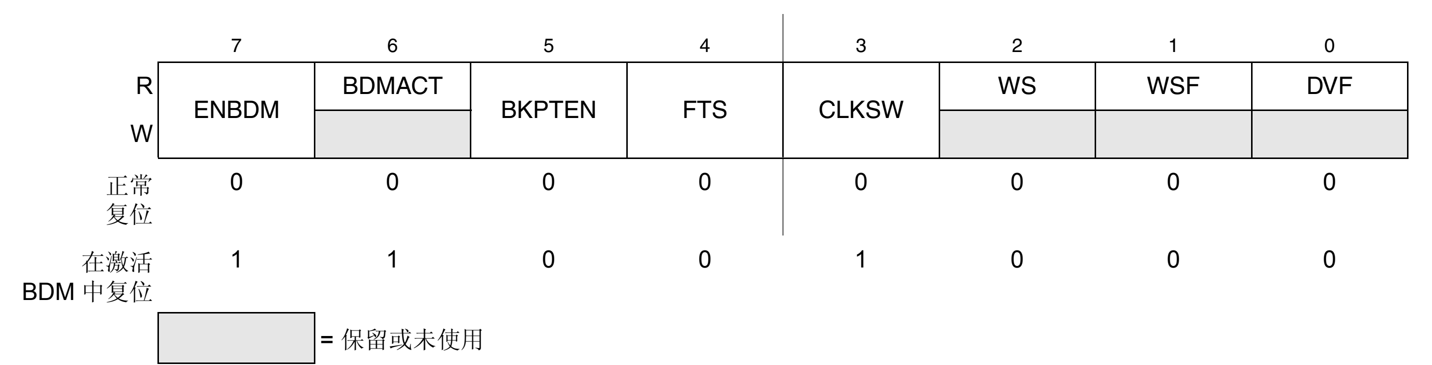


图 16-5. BDC 状态和控制寄存器 (BDCSCR)

表 16-2. BDCSCR 寄存器字段描述

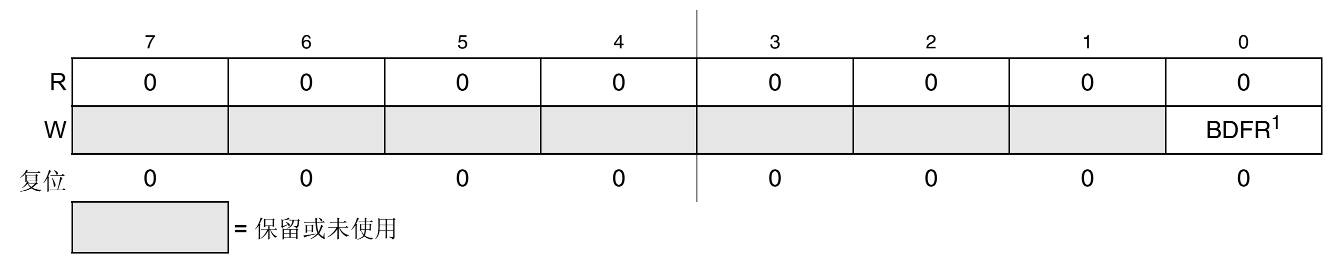
字段	描述
7 ENBDM	**激活 BDM (允许激活后台调试模式)** — 一般而言，这个位在调试开始后不久，或只要调试主机复位目标，由调试主机写为 1，并保留 1，直到通过正常的复位清除它。 0 BDM 不能激活 (非侵入式命令仍然被允许) 1 BDM 可以激活，允许激活后台模式命令
6 BDMACT	**后台调试模式激活状态** — 这是只读状态位。 0 BDM 未激活 (用户应用程序运行) 1 BDM 激活并等待串行命令
5 BKPTEN	**BDC 断点激活** — 如果这个位清零，BDC 断点处于处活状态，FTS (强制标签选择) 控制位和 BDCBKPT 匹配寄存器被忽略。 0 BDC 断点禁止 1 BDC 断点激活
4 FTS	**强制 / 标签选择** — 当 FTS = 1, 只要 CPU 地址总线匹配 BDCBKPT 匹配寄存器，则请求断点。当 FTS = 0, CPU 地址总线与 BDCBKPT 寄存器之间的匹配会造成获取的操作码被标记。如果标记的操作码到达指令队列的末端， CPU 则进入激活后台模式，而不是执行标记的操作码。 0 在断点地址标记操作码，如果 CPU 试图执行该指令，则进入激活后台模式 1 断点匹配强制在下一个指令边界进入激活后台模式 (地址不必是操作码)
3 CLKSW	**选择 BDC 通信时钟的源** — CLKSW 默认 0，选择其它 BDC 时钟源。 0 其它 BDC 时钟源 1 MCU 总线时钟表
2 WS	**等待或停止状态** — 当目标 CPU 处于等待或停止状态时，大多数 BDC 命令不起作用。但是可以用后台命令来强制目标 CPU 从等待或停止状态进入激活后台模式，这样所有 BDC 命令都可以起作用。只要主机强制目标 MCU 进入激活后台调试模式 , 主机应该发出 READ_STATUS 命令，在偿试其它 BDC 命令前，检查 BDMACT = 1 。 0 目标 CPU 运行用户应用代码，或处于激活后台调试模式 (当后台激活时，它不处于等待或停止模式) 1 目标 CPU 处于等待或停止模式，或者后台命令用来将其从等待或停止状态改变为激活后台调试模式
1 WSF	**等待或停止失败状态** — 如果这存储器存取命令因目标 CPU 在大约相同时间执行等待或停止指令而失败，则设置这个状态位。通常的恢复策略是发出后台命令，从等待或停止模式进入激活后台模式，重复失败的命令，然后返回到用户程序。 (一般地，主机应该恢复 CPU 寄存器 ，准备值，重新执行等待或停止指命。) 0 存储器存取与等待或停止指令不冲突 1 存储器存取命令失败，因为 CPU 已进入等待或停止模式
0 DVF	**数据有效失败状态** — 这个状态位没有在 MC9S08DZ60 系列中使用，因为它没有慢存取存储器。 0 存储器存取与慢存储器接入不冲突 1 存储器存取命令失败，因为 CPU 没有完成慢存储器接入

16.4.1.2 BDC 断点匹配寄存器 (BDCBKPT)

6 位 寄存器保留 BDC 中的硬件断点的地址。BDCSCR 中的 BKPTEN 和 FTS 控制位用来使能和配置断点逻辑。专门的串行 BDC 命令 (READ_BKPT 和 WRITE_BKPT) 用来读和写 BDCBKPT 寄存器，但是用户程序不能存取它，因为它不位于 MCU 的普通存储器映射空间中。当目标 MCU 处于激活后台调试模式时，断点一般在运行用户应用程序前设置。关于建立和使用 BDC 中的硬件断点逻辑的更多信息，请参见 16.2.4 节 BDC 硬件断点。

16.4.2 系统后台调试强制复位寄存器 (SBDFR)

这个寄存器包含单个只写控制位。必须要用一个串行后台模式命令，如 WRITE_BYTE，来写 SBDFR。从用户程序写该寄存器的偿试被忽略。读总是返回 0x00。



	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								BDFR[1]
复位	0	0	0	0	0	0	0	0

= 保留或未使用

[1] BDFR 只有通过串行后台模式调试命令才可写，不能通过用户程序来写。

图 16-6. 系统后台调试强制复位寄存器 (SBDFR)

表 16-3. 寄存器字段描述

字段	描述
0 BDFR	后台调试强制复位 — 一系列激活后台模式命令，如 WRITE_BYTE 等，允许外部调试主机强制目标系统复位。将 1 写到这个位，强制 MCU 复位。这个位 不能从用户程序写。

16.4.3 DBG 寄存器和控制位

这个调试模块包括 9 个字节的寄存器空间，用于三个 16 位寄存器和三个 8 位控制和状态寄存器。这些寄存器位于存储器空间的高地址空间中，这样它们可以存取正常的应用程序。普通用户应用程序几乎从不接入这些寄存器，除了使用断点逻辑的 ROM patching 机制。

16.4.3.1 调试比较器 A 高寄存器 (DBGCAH)

这个寄存器包含比较器 A 的高 8 位的比较值位。在复位时，这个寄存器被强制设置为 0x00，可以随时被读或写，除非 ARM = 1。

16.4.3.2 调试比较器 A 低寄存器 (DBGCAL)

这个寄存器包含比较器 A 的低 8 位的比较值位。在复位时，这个寄存器被强制设置为 0x00，可以随时被读或写，除非 ARM = 1。

16.4.3.3 调试比较器 B 高寄存器 (DBGCBH)

这个寄存器包含比较器 B 的高 8 位的比较值位。在复位时，这个寄存器被强制设置为 0x00，可以随时被读或写，除非 ARM = 1。

16.4.3.4 调试比较器 B 低寄存器 (DBGCBL)

这个寄存器包含比较器 B 的低 8 位的比较值位。在复位时，这个寄存器被强制设置为 0x00，可以随时被读或写，除非 ARM = 1。

16.4.3.5 调试 FIFO 高寄存器 (DBGFH)

这个寄存器提供对 FIFO 的高 8 位的只读接入。写到这个寄存器没有意义或无效果。在纯事件触发模式中，FIFO 只将数据存储在每个 FIFO 字的低字节，因此这个寄存器不能使用，将读 0x00。

读 DBGFH 不会导致 FIFO 移动到下一个字。当从 FIFO 中读出 16 位字时，在读 DBGFL 前先读 DBGFH ，因为读 DBGFL 会导致 FIFO 先于下个字的信息。

16.4.3.6 调试 FIFO 低寄存器 (DBGFL)

这个寄存器提供对 FIFO 的低 8 位的只读存取。写到这个寄存器没有意义或无效果。

读 DBGFL 会导致 FIFO 移动到下一个字的信息。当调试模块以纯事件模式运行时，只有 8 位数据存储在 FIFO （每个 FIFO 字的高字节部分没有使用）。当从 FIFO 中读出 8 位字时，只需重复地读 BDGFL，从 FIFO 中获得数据的连续的字节。在这种情况下，没有必要读 DBGFH。

当 FIFO 仍然打开时 （打开后，但 FIFO 充满或 ARMF 被清除前）不要试图从其中读数据，因为在 DBGL 读取过程中， FIFO 不能进一步操作。这可以干扰正常的 FIFO 的读取顺序。

在调试器没有打开的情况下读 会使最近获取的操作码的地址存储到 FIFO 中的最后的位置。读取 DBGFL，然后定期 DBGFL，外部主机软件可以开发程序执行的概况。在对 FIFO 进行八次读取后，第九次读取将返回第一次读取结果的信息。要使用分析功能，则需要读取 FIFO 八次，且不使用启动顺序的数据，然后开始使用数据来获取已执行地址的延迟概貌。存储在 FIFO 中的关于 DBGFL (且 FIFO 没有打开) 读取的信息就是最近所获操作码的地址。

16.4.3.7 调试控制寄存器 (DBGC)

这个寄存器可以在任何时间读或写。

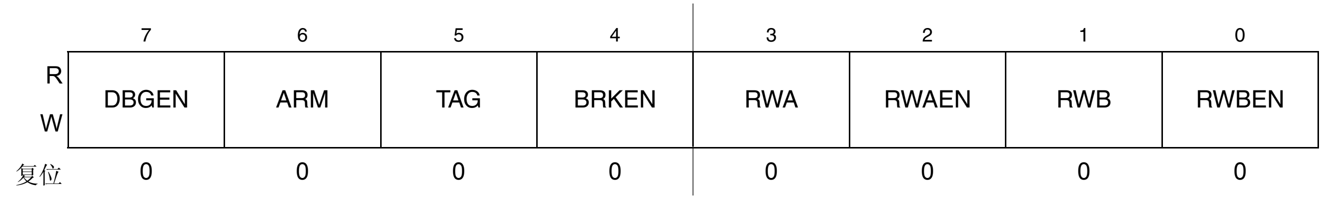


图 16-7. 调试控制寄存器 (DBGC)

表 16-4. DBGC 寄存器字段描述

字段	描述
7 DBGEN	调试模块启用 — 来用启用调试模块。 DBGEN 不能设置为 1，如果 MCU 是安全的。 0 DBG 禁用 1 DBG 启用
6 ARM	打开控制 — 控制调试器是否在 FIFO 中比较和存储信息。采用写操作来设置该位 (和 ARMF) ，完成调试运行就是自动清除它。将 ARM 或 DBGEN 写为 0，可以停止任何调试运行。 0 调试器没有打开 1 调试器被打开
5 TAG	标记 / 强制选择 — 控制送到 CPU 的中断请求是否为标签或强制型请求。如果 BRKEN = 0，这个位就没有意义或无效。 0 CPU 中断请求作为强制型请求 1 CPU 中断请求作为标签型请求
4 BRKEN	中断启用 — 控制触发事件是否向 CPU 生成中断请求。触发事件可以使信息存储在 FIFO 中而不必向 CP 生成中断请求。对于结束跟踪，如果比较器 (s) 和 R/W 满足触发条件，则发出 CPU 中断请求。对于起始跟踪，则当 FIFO 满时发出 CPU 中断请求。 TRGSEL 不影响 CPU 中断请求的定时。 0 CPU 不断请求未启用 1 触发器触发向 CPU 发出中断请求
3 RWA	比较器 A 的 R/W 比较值 — 当 RWAEN = 1, 这个位确定是否用读或写接入来鉴定比较器 A，当 RWAEN = 0, , RWA 和 R/W 信号不影响比较器 A。 0 比较器 A 只在写周期上匹配 1 比较器 A 只在读周期上匹配
2 RWAEN	**启用比较器 A 的 R/W** — 控制比较器 A 的匹配是否考虑这个水平的 R/W 。 0 R/W 未用在比较 A 中 1 R/W 用在比较 A 中
1 RWB	比较器 B 的 R/W 比较值 — 当 RWBEN = 1, 这个位确定是否用读或写接入来鉴定比较器 B。当 RWBEN = 0, RWA 和 R/W 信号不影响比较器 B。 0 比较器 B 只在写周期上匹配 1 比较器 B 只在读周期上匹配
0 RWBEN	**启用比较器 B 的 R/W B** — 控制比较器 B 的匹配是否考虑这个水平的 R/W 。 0 R/W 未用在比较 B 中 1 R/W 用在比较 B 中

16.4.3.8 调试触发寄存器 (DBGT)

这个寄存器在任何时候都可以读，但是只有当 ARM = 0 时才可以写，除非位 4 和位 5 硬件线与至 0。

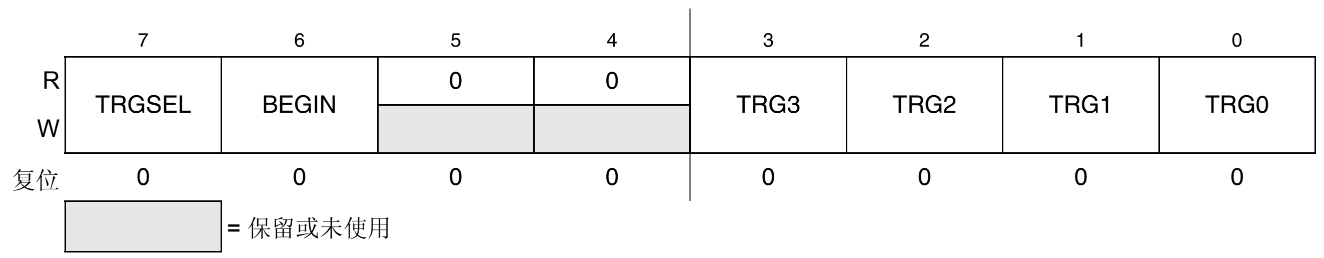


图 16-8. 调试触发寄存器 (DBGT)

表 16-5. DBGT 寄存器字段描述

字段	描述
7 TRGSEL	触发类型 — 控制比较器 A 和 B 的匹配输入是否与调试模块中的操作码跟踪逻辑匹配。如果 TRGSEL 已设置，比较器 A 或 B 的匹配信号必须通过操作码跟踪逻辑传播，如果匹配地址的操作码实际已执行，则只有触发事件发送到 FIFO 逻辑。 0 存取比较地址时触发 (强制) 1 如果比较地址的操作码已执行 (标签)，则触发
6 BEGIN	开始 / 结束触发选择 — 控制 FIFO 在触发时开始填充还是以循环形式填充直到触发结束信息的捕获。在纯事件触发模式中，忽略这个位，所有调试运用都假定为起始跟踪。 0 数据存储在 FIFO，直到触发 （结束跟踪） 1 触发启动数据存储 (起始跟踪)
3:0 TRG[3:0]	选择触发模式 — 选择下面 9 个触发模式中的一个。 0000 只有 A 0001 A 或 B 0010 A 然后 B 0011 只有事件 B (存储数据) 0100 A 然后只有事件 B (存储数据) 0101 A 和 B 数据 (满模式) 0110 A 和非 B 数据 (满模式) 0111 I 范围内：A ≤ 地址≤ B 1000 范围外：地址 < A 或 地址 > B 1001 – 1111 (无触发)

16.4.3.9 调试状态寄存器 (DBGS)

这是一个只读状态寄存器。

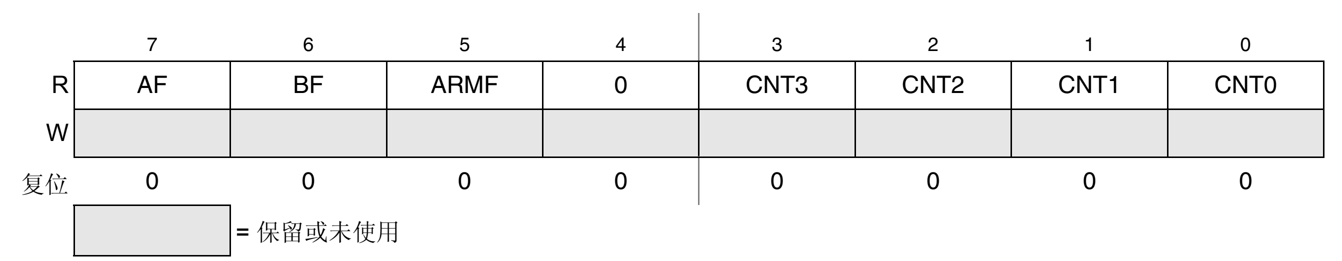


图 16-9. 调试状态寄存器 (DBGS)

表 16-6. DBGS 寄存器字段描述

字段	描述
7 AF	**触发匹配 A 标记 —** 在调试运行开始时请除 AF，指示武装后是否满足触发 匹配 A 条件。 0 比较器 A 未匹配 1 比较器 A 匹配
6 BF	**触发匹配 B 标记 —** 在调试运行开始时请除 BF，指示武装后是否满足触发 匹配 B 条件。 0 比较器 B 未匹配 1 比较器 B 匹配
5 ARMF	**打开标记 —** 当 DBGEN=1 时，这个位为 DBGC 中 ARM 的只读镜像。将 DBGC 中的 ARM 控制位写为 1 （当 DBGEN = 1）可设置该位，在调试运行结束时自动清除它。当 FIFO 为满时 （始起跟踪），或当探测到触发事件时 （结束跟踪），调度运行完成。将 DBGC 中的 ARM 或 DBGEN 写为 0，可以人工停止调试运行 。 0 调试器没有打开 1 调试器被打开
3:0 CNT[3:0]	**FIFO 有效计数 —** 这些位在调试运行开始时清除，指示调试运行结束时 FIFO 中的有效数据的字数。当数据大 FIFOO 中读出时， CNT 中的值不减少。当信息从 FIFO 中读出时，外部调试主机负责计数的跟踪。 0000 FIFO 中的有效字数 = 无有效数据 0001 FIFO 中的有效字数 = 1 0010 FIFO 中的有效字数 = 2 0011 FIFO 中的有效字数 = 3 0100 FIFO 中的有效字数 = 4 0101 FIFO 中的有效字数 = 5 0110 FIFO 中的有效字数 = 6 0111 FIFO 中的有效字数 = 7 1000 FIFO 中的有效字数 = 8

附录 A
电气特性和时序规范

A.1 引言

本章包含了微控制器的电气和时序规范。

A.2 参数分类

这个附录中显示的电气参数通过多种方式对其正确性进行保证。为了让客户更好地理解，我们进行了如下分类，并对表中的参数进行了相应地标注：

表 A-1. 参数分类

P	在生产测试时对各个器件测试得到。
C	这些参数通过设计特性表征来实现的，方法是从统计上在不同条件下测量相当数量的样本得到。
T	除非另有说明，参数是通过确定典型条件下典型设备的少量样本测试得到。典型列中显示的所有数值都在这个范畴以内。
D	主要通过仿真得到。

注意

参数表“C”栏中显示了相应的分类。

A.3 绝对最大额定值

绝对最大额定值是应力额定值，不能保证在绝对最大值下正常工作。应力超过 表 A-2 中规定的极限时，会影响器件可靠性或对器件造成永久损害。关于正常工作的条件，请参见本节的其他表格。

器件包含了防止高静态电压或电场损害的保护电路，但是我们仍建议采取正常预防措施以避免高于最大额定电压的电压进入该高阻抗电路。未使用的输入连接到一个合适的逻辑电压 （例如，V_{SS} 或 V_{DD}），可以增强操作的可靠性。

表 A-2. 最大绝对额定值

参数	符号	值	单位
电源电压	V_{DD}	–0.3 ～ 5.8	V
输入电压	V_{In}	– 0.3 ～ V_{DD} + 0.3	V
瞬时最大电流 单引脚极限 （适用于所有端口引脚）[1] , [2] , [3]	I_D	± 25	mA
V_{DD} 中最大电流	I_{DD}	120	mA
存储温度	T_{stg}	–55 ～ 150	°C
最大结点温度	T_J	150	°C

[1] 输入必须是限定为指定值的电流。要确定所需的电流限定电阻值，需要先计算用于正 (V_{DD}) 和负 (V_{SS}) 钳位电压的电阻值，然后使用两个电阻值中的较大者。

[2] 所有功能性非电源引脚内部均钳位在 V_{SS} 和 V_{DD}。

[3] 在瞬时和操作最大电流条件下，电源必须维持在 V_{DD} 工作范围内。如果正注入电流 ($V_{In} > V_{DD}$) 大于 I_{DD}，则注入电流就可能超出 V_{DD} ，并导致外部电源不可调控。确保外部 V_{DD} 负载分流大于最大注入电流的电流。当 MCU 不消耗功率时，存在最大的风险，例如，如果当前无系统时钟，或者如果时钟速率非常低，这都会降低总功耗。 .

A.4 热特性

本小节介绍了工作温度范围、功耗和封装热阻的信息。I/O 引脚的功耗通常比片上逻辑的功耗小，它是由用户决定的而不是 MCU 设计控制的。若要在功耗计算中考虑 $P_{I/O}$，需要确定实际引脚电压和 V_{SS} 或 V_{DD} 之间的误差，然后乘以每个 I/O 引脚的电流。除了特别的大引脚电流 （大负载），引脚电压与 V_{SS} 或 V_{DD} 的误差非常小，可以忽略不计。

表 A-3. 热特性

参数		符号	值	单位
工作温度范围 （已封装）		T_A	T_L to T_H –40 to 125	°C
热阻 [1,2,3,4]		θ_{JA}		°C/W
64 引脚 QFP	1s		57	
	2s2p		43	
64 引脚 LQFP	1s		69	
	2s2p		54	
48 引脚 QFN	1s		84	
	2s2p		27	
44 引脚 LQFP	1s		73	
	2s2p		56	
32 引脚 LQFP	1s		85	
	2s2p		56	

[1] 结点温度是晶元尺寸、片上功耗、封装热阻、安装点 （卡板）的温度、周围温度、气流、板上其它组件的功耗以及主板热阻的函数。

[2] 结与环境的自然对流

[3] 1s - 单层卡板，一个信号层

[4] 2s2p - 四层卡板，2 个信号层和 2 个电源层

平均芯片接面温度 (T_J) (℃) 可以用以下等式计算出来：

$$T_J = T_A + (P_D \times \theta_{JA})$$ *等式 A-1*

其中：

T_A = 周围温度 , ℃
θ_{JA} = 封装热阻，结点至周围环境，℃ /W
$P_D = P_{int} + P_{I/O}$
$P_{int} = I_{DD} \times V_{DD}$，瓦特 — 芯片内部功率
$P_{I/O}$ = 输入和输出引脚上的功耗 — 由用户决定

在大多数应用中， $P_{I/O} << P_{int}$ ，可以忽略不计。P_D 和 T_J （如果忽略 $P_{I/O}$）的近似关系是：

$$P_D = K \div (T_J + 273°C)$$ *等式 A-2*

解等式 A-1 和等式 A-2 ，得 K：:

$$K = P_D \times (T_A + 273°C) + \theta_{JA} \times (P_D)^2$$ *等式 A-3*

其中 K 是与特定部分相关的常量。已知 T_A，通过测量 P_D （在平衡时），可以在等式 A-3 中得出 K。使用 K 值，来解等式 A-1 和等式 A-2 ，可以得到任何 T_A 值的 P_D 和 T_J。

A.5 ESD 保护和抗闭锁方法

虽然静电放电对这些器件造成的伤害远没有对早期的 CMOS 电路那么平常，但是仍需要采用一些预防手段来防止静电。质量测试用来确保当这些器件暴露在合理的静电中，不会被永久性破坏。

整个 ESD 测试都符合汽车集成电路的 AEC-Q100 应力测试认证和非汽车集成电路的 JEDEC 标准。在 ESD 质量测试过程中，要执行人体放电模式 （HBM）、机器放电模式和充电器件模式 （CDM）的 ESD 应力测试。

如果暴露在 ESD 脉冲后，器件不再符合器件规范，那么就认定器件测试失败。在进行室温测试后，还要在高温下，根据每个适用的器件规范进行完整的 DC 参数及功能测试，除非设备规范中另有说明。 .

表 A-4. ESD 和闭锁测试条件

模式	描述	符号	值	单位
人体放电	串联电阻	R1	1500	Ω
	存储电容	C	100	pF
	每引脚脉冲数	—	3	
机器	串联电阻	R1	0	Ω
	存储电容	C	200	pF
	每引脚脉冲数	—	3	
闭锁	最小输入电压限制		–2.5	V
	最大输入电压限制		7.5	V

表 A-5. ESD 和闭锁保护特性

编号	C	参数	符号	最小值	典型值[1]	最大值
1	C	人体模式 (HBM)	V_{HBM}	±2000	—	V
2	C	机器模式 (MM)	V_{MM}	±200	—	V
3	C	充电器件模式 (CDM)	V_{CDM}	±500	—	V
4	C	T_A = 125° C 时的闭锁电流	I_{LAT}	±100	—	mA

[1] 除非另有说明，典型值是温度在 25° C 时的数据 。

A.6 DC 特性

本小节介绍了各种操作模式中的电源要求、 I/O 引脚特性及电源电流信息。

表 A-6. DC 特性

编号	C	参数	符号	最小值	典型值[1]	最大值	单位
1	P	高压输出 — 低驱 (PTxDSn = 0) 5 V, I_{Load} = –2 mA 3 V, I_{Load} = –0.6 mA 5 V, I_{Load} = –0.4 mA 3 V, I_{Load} = –0.24 mA	V_{OH}	 V_{DD} – 1.5 V_{DD} – 1.5 V_{DD} – 0.8 V_{DD} – 0.8	 — — — —	 — — — —	V
		高压输出 — 高驱 (PTxDSn = 1) 5 V, I_{Load} = –10 mA 3 V, I_{Load} = –3 mA 5 V, I_{Load} = –2 mA 3 V, I_{Load} = –0.4 mA		 V_{DD} – 1.5 V_{DD} – 1.5 V_{DD} – 0.8 V_{DD} – 0.8	 — — — —	 — — — —	
2	P	低压输出 — 低驱 (PTxDSn = 0) 5 V, I_{Load} = 2 mA 3 V, I_{Load} = 0.6 mA 5 V, I_{Load} = 0.4 mA 3 V, I_{Load} = 0.24 mA	V_{OL}	 1.5 1.5 0.8 0.8	 — — — —	 — — — —	V
		低压输出 — 高驱 (PTxDSn = 1) 5 V, I_{Load} = 10 mA 3 V, I_{Load} = 3 mA 5 V, I_{Load} = 2 mA 3 V, I_{Load} = 0.4 mA		 1.5 1.5 0.8 0.8	 — — — —	 — — — —	
3	P	高电流输出 — 所有端口的最大总 I_{OH} 5V 3V	I_{OHT}	 — —	 — —	 100 60	mA
4	P	低电流输出 — 所有端口的最大总 I_{OL} 5V 3V	I_{OLT}	 — —	 — —	 100 60	mA
5	P	高压输入；所有数字输入 2.7v ≤ V_{DD} 4.5v	V_{IH}	0.70xV_{DD}	—	—	V
		4.5v ≤ V_{DD} ≤ 5.5v	V_{IH}	0.65xV_{DD}	—	—	
6	P	低压输入 ; 所有数字输入	V_{IL}	—	—	0.35 x V_{DD}	
7	P	输入滞后，所有数字输入	V_{hys}	0.06 x V_{DD}			mV
8	P	输入漏电流，仅针对所有输入管脚[2]	$\|I_{In}\|$	—	0.1	1	μA

表 A-6. DC 特性（续）

编号	C	参数	符号	最小值	典型值[1]	最大值	单位
9	P	高阻抗 (关闭状态) 漏电流 [2]	$\|I_{OZ}\|$	—	0.1	1	μA
10	P	内部上拉电阻[3]	R_{PU}	20	45	65	kΩ
11	P	内部下拉电阻[4]	R_{PD}	20	45	65	kΩ
12	C	输入电容；所有非电源引脚	C_{In}	—	—	8	pF
13	D	RAM 保持电压	V_{RAM}	—	0.6	1.0	V
14	P	POR rearm 电压	V_{POR}	0.9	1.4	2.0	V
15	D	POR rearm 时间	t_{POR}	10	—	—	ms
16	P	低压探测阈值 — 高量程 V_{DD} 下降沿 V_{DD} 上升沿	V_{LVDH}	4.2 4.3	4.3 4.4	4.4 4.5	V
17	P	低压探测阈值 — 低量程 V_{DD} 下降沿 V_{DD} 上升沿	V_{LVDL}	2.48 2.54	2.56 2.62	2.64 2.7	V
18	P	低压报警阈值 — 高量程 V_{DD} 下降沿 V_{DD} 上升沿	V_{LVWH}	4.2 4.3	4.3 4.4	4.4 4.5	V
19	P	低压报警阈值 — 低量程 V_{DD} 下降 V_{DD} 上升	V_{LVWL}	2.48 2.54	2.56 2.62	2.64 2.7	V
20	P	低压禁止复位 / 恢复滞后 5V 3V	V_{hys}	— —	100 60	— —	mV

[1] 除非另有说明，典型值是温度在 25° C 时的数据 。
[2] $V_{In} = V_{DD}$ 或 V_{SS} 时的测量值。
[3] $V_{In} = V_{SS}$ 时的测量值。
[4] $V_{In} = V_{DD}$ 时的测量值。

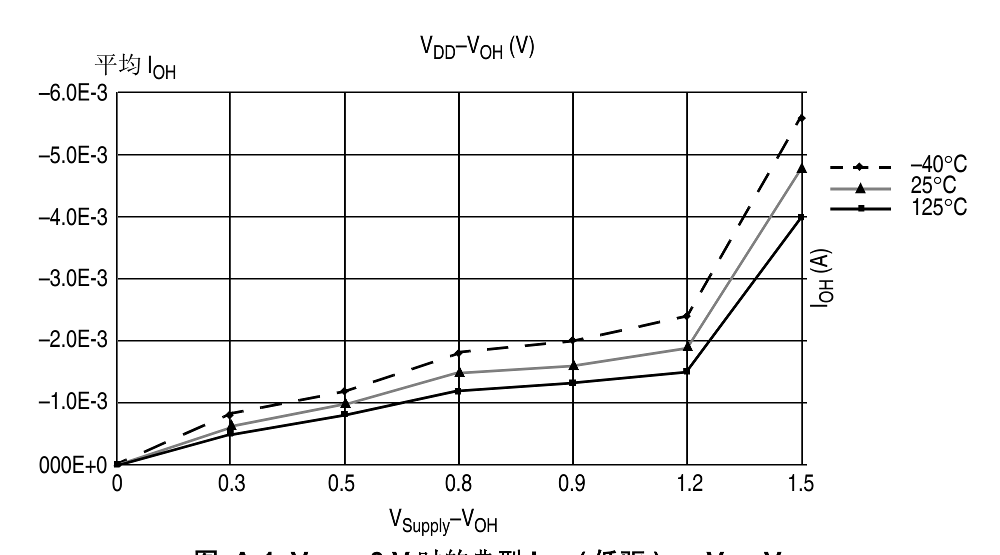


图 A-1. V_{DD} = 3 V 时的典型 I_{OH} (低驱) vs V_{DD}-V_{OH}

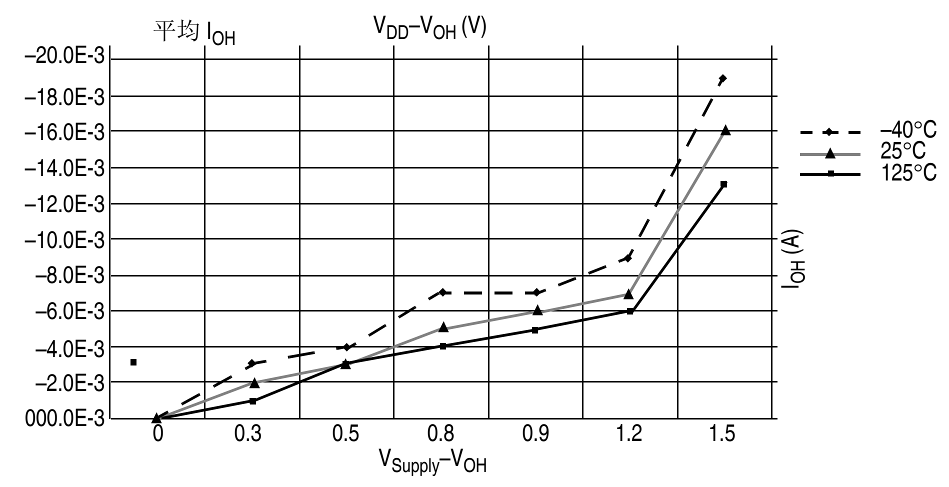


图 A-2. V_{DD} = 3 V 时的典型 I_{OH} (高驱) vs V_{DD}-V_{OH}

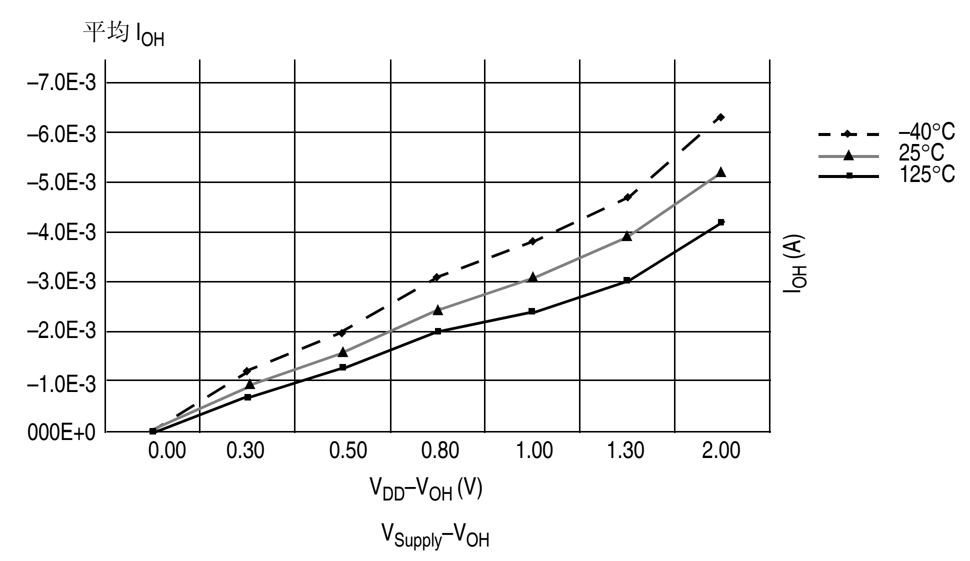


图 A-3. V_{DD} = 5 V 时的典型 I_{OH} (低驱) vs V_{DD}-V_{OH}

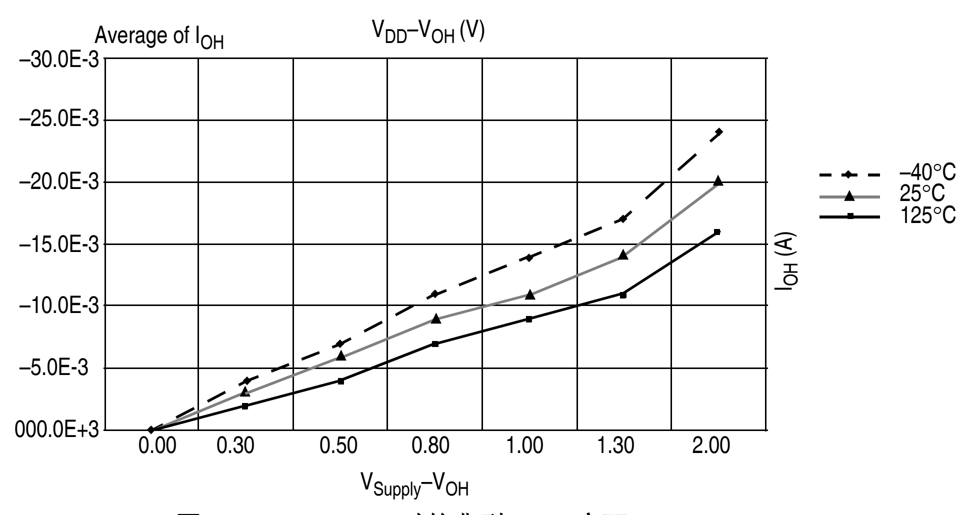


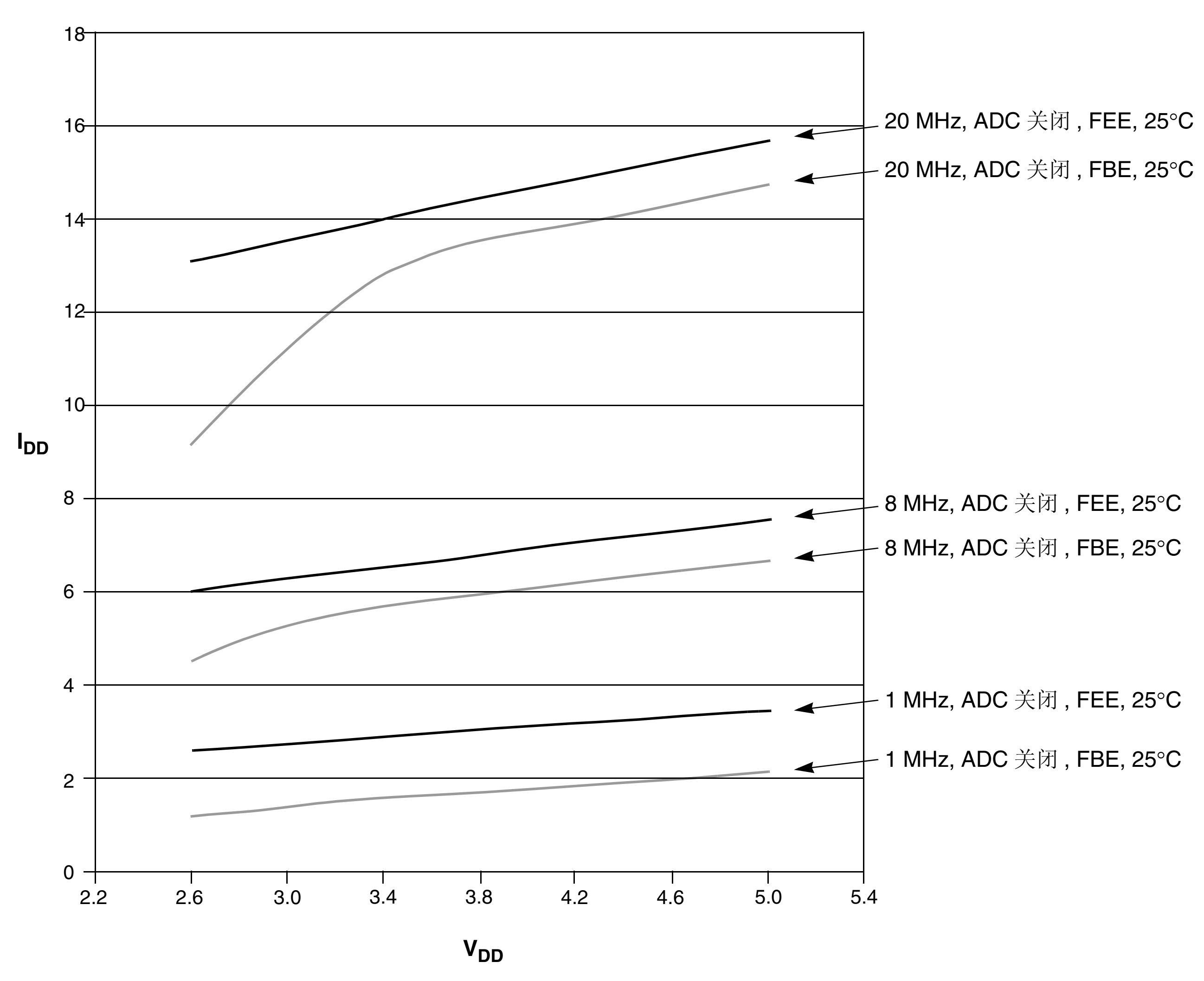
图 A-4. V_{DD} = 5 V 时的典型 I_{OH} (高驱) vs V_{DD}-V_{OH}

A.7 电源电流特性

表 A-7. 电源电流特性

编号	C	参数	符号	V_{DD} (V)	典型值[1]	最大值[2]	单位	温度 (°C)
1	C	当 (CPU 时钟 = 2 MHz, f_{Bus} = 1 MHz) 时测量的运行电源电流[3]	RI_{DD}	5	0.750	0.950[4]	mA	–40 to 125°C
				3	0.570	0.770		
2	C	当 (CPU 时钟 = 16 MHz, f_{Bus} = 8 MHz) 时测量的运行电源电流[5]	RI_{DD}	5	4.9	5.10[6]	mA	–40 to 125°C
				3	3.5	3.70		
3	P	停止 2 模式电源电流	$S2I_{DD}$	5	0.900	18.0 60[4]	μA	0 to 70°C –40 to 85°C –40 to 125°C
				3	0.720	17.0 50	μA	0 to 70°C –40 to 85°C –40 to 125°C
4	P	停止 3 模式电源电流	$S3I_{DD}$	5	0.975	20.0 90[4]	μA	0 to 70°C –40 to 85°C –40 to 125°C
				3	0.825	19.0 85	μA	0 to 70°C –40 to 85°C –40 to 125°C
5	C	停止 2 或停止 3[7] 下 RTI 消耗电流	$S23I_{DDRTI}$	5	300	500 500	nA	–40 to 85°C –40 to 125°C
				3	300	500 500	nA	–40 to 85°C –40 to 125°C
6	C	停止 3 的 LVD 消耗电流 (LVDE = LVDSE = 1)	$S3I_{DDLVD}$	5	110	180	μA	–40 to 85°C –40 to 125°C
				3	90	160	μA	–40 to 85°C –40 to 125°C
7	C	停止 3 用于振荡器启用的消耗电流[8] (OSCSTEN =1)	$S3I_{DDOSC}$	5,3	5	8	μA	–40 to 85°C –40 to 125°C

[1] 除非另有说明，典型值是在 25° C 时的特性值 。有关电压 / 温度的典型曲线，参见 图 A-5 至图 A-7。
[2] 所给出的值是完成特性表征前的初始估算值。
[3] 所有模块，除 ADC，均为激活状态，ICG 配置给 FBE，不包含端口引脚上的任何 DC 负载。
[4] 每个单元均测试这个参数。最大值栏中的所有其它值均通过特性表征验证。
[5] 所有模块，除 ADC，均为激活状态，ICG 配置给 FBE，不包含端口引脚上的任何 DC 负载。
[6] 每个单元均测试这个参数。最大值 （Max）栏中的所有其它值均通过特性表征验证。
[7] 大多数客户会发现他们可以使用停止 2 或停止 3 模式自动唤醒，而不必使用电流更高的等待模式。等待模式的典型值是 560 μA ，在电压为 3 V，f_{Bus} = 1 MHz 时。
[8] 所给的值基于以下条件：低量程操作 (RANGE = 0) ， 32.768 kHz 晶振，低功率模式 (HGO = 0)，时钟监控关闭 (LOCD = 1)。



注意：外部时钟是函数发生器提供的方波。对于 FEE 模式，外部参考频率是 4 MHz

图 A-5. FBE 和 FEE 模式的典型运行 I_{DD}，I_{DD} vs. V_{DD}

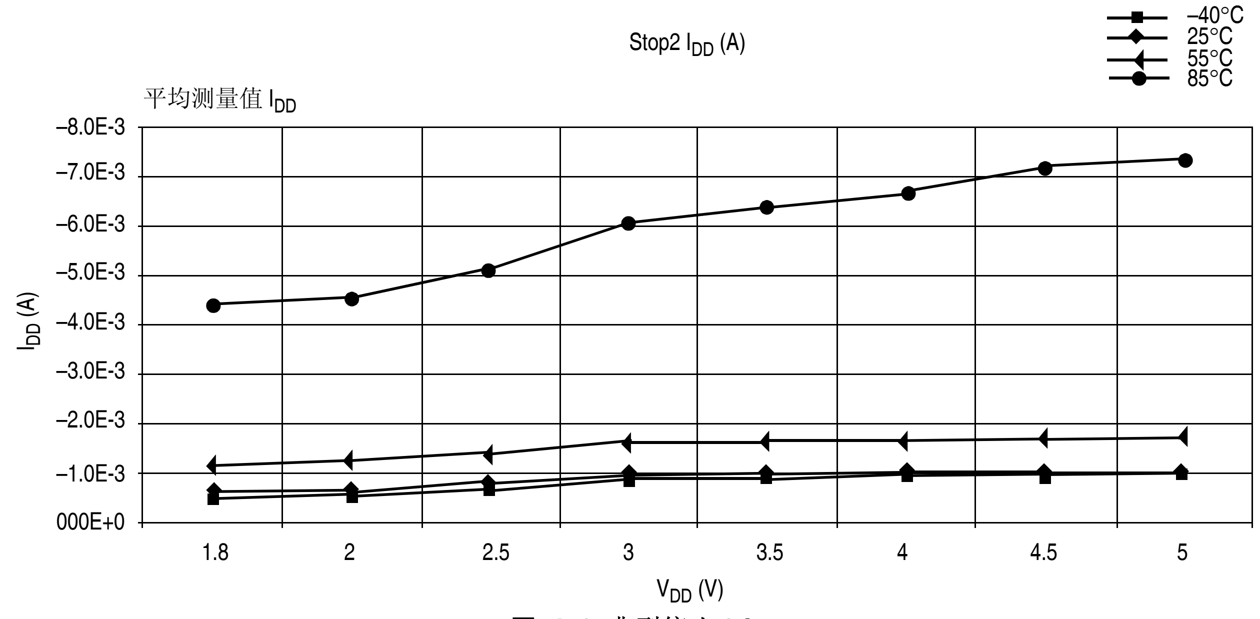




图 A-6. 典型停止 **2 I_{DD}**

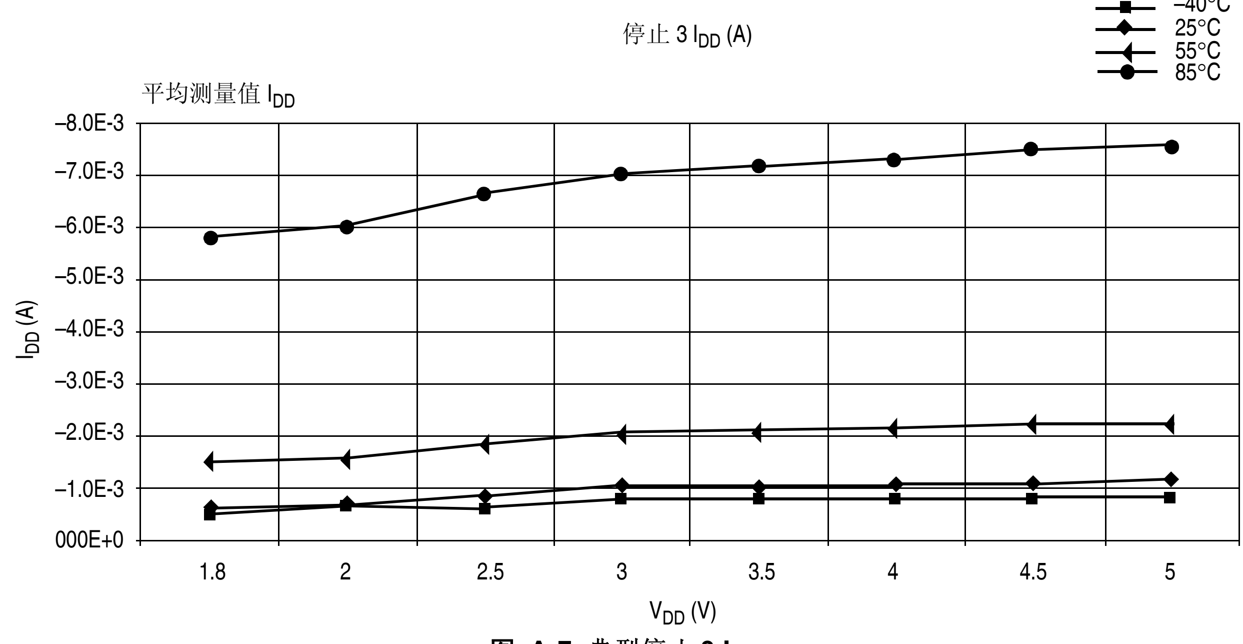


图 A-7. 典型停止 **3 I_{DD}**

A.8 ADC 特性

表 A-8. 5 V 10 位 ADC 操作条件

特性	条件	符号	最小值	典型值[1]	最大值	单位
电源电压	绝对	V_{DDAD}	2.7	—	5.5	V
	基于 V_{DD} 的变量 (V_{DD}-V_{DDAD})[2]	ΔV_{DDAD}	–100	0	100	mV
接地电压	基于 V_{SS} 的变量 (V_{SS}-V_{SSAD})[2]	ΔV_{SSAD}	–100	0	100	mV
参考电压 高		V_{REFH}	2.7	V_{DDAD}	V_{DDAD}	V
参考电压 低		V_{REFL}	V_{SSAD}	V_{SSAD}	V_{SSAD}	V
电源电流	停止，复位，模块关闭	I_{DDAD}	—	0.011	1	μA
输入电压		V_{ADIN}	V_{REFL}	—	V_{REFH}	V
输入电容		C_{ADIN}	—	4.5	5.5	pF
输入电阻		R_{ADIN}	—	3	5	kΩ
模拟信号源电阻 MCU 外部源	10 位模式 f_{ADCK} > 4MHz f_{ADCK} < 4MHz	R_{AS}	 — —	 — —	 5 10	kΩ
	8 位模式 (所有有效 f_{ADCK})		—	—	10	
ADC 转换时钟频率	高速 (ADLPC = 0)	f_{ADCK}	0.4	—	8.0	MHz
	低功率 (ADLPC = 1)		0.4	—	4.0	
温度传感器 斜率	-40°C– 25°C	m	—	3.266	—	mV/°C
	25°C– 125°C			3.638	—	
温度传感器 电压	25°C	V_{TEMP25}	—	1.396	—	V

[1] 典型值假设 在 V_{DDAD} = 5.0 V, Temp = 25°C，f_{ADCK} = 1.0MHz 时测得，除非另有其他说明。典型值仅作参考用，在生产中未测试。

[2] dc 电势差。

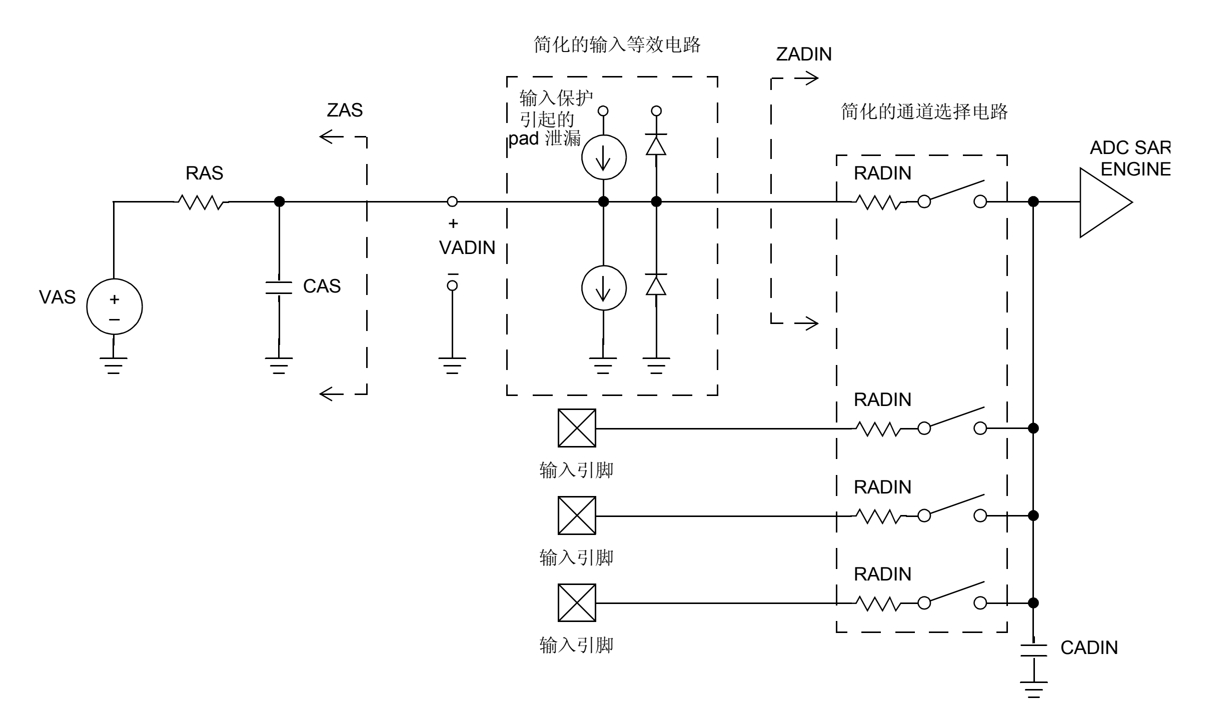


图 A-8. ADC 输入抗阻等效示意图

表 A-9. 5 V 10 位 ADC 特性 (V_{REFH} = V_{DDAD}, V_{REFL} = V_{SSAD})

特性	条件	C	条件	符号	最小值	典型值[1]	最大值
电源电流 ADLPC = 1 ADLSMP = 1 ADCO = 1		T	I_{DDAD}	—	133	—	μA
电源电流 ADLPC = 1 ADLSMP = 0 ADCO = 1		T	I_{DDAD}	—	218	—	μA
电源电流 ADLPC = 0 ADLSMP = 1 ADCO = 1		T	I_{DDAD}	—	327	—	μA
电源电流 ADLPC = 0 ADLSMP = 0 ADCO = 1		T	I_{DDAD}	—	582	—	μA
	$V_{DDAD} \leq 5.5$ V	P		—	—	1	mA
ADC 异步时钟源 t_{ADACK} = $1/f_{ADACK}$	高速 (ADLPC = 0)	P	f_{ADACK}	2	3.3	5	MHz
	低功率 (ADLPC = 1)			1.25	2	3.3	
转换时间 (包括采样时间)	短采样 (ADLSMP = 0)	P	t_{ADC}	—	20	—	ADCK cycles
	长采样 (ADLSMP = 1)			—	40	—	
采样时间	短采样 (ADLSMP = 0)	P	t_{ADS}	—	3.5	—	ADCK cycles
	长采样 (ADLSMP = 1)			—	23.5	—	
未调整误差总数 包括量化	10 位模式	P	E_{TUE}	—	±1	±2.5	LSB[2]
	8 位模式			—	±0.5	±1.0	
差分非线性	10 位模式	P	DNL	—	±0.5	±1.0	LSB[2]
	8 位模式			—	±0.3	±0.5	
	保证无变化，无缺失代码						
积分非线性	10 位模式	C	INL	—	±0.5	±1.0	LSB[2]
	8 位模式			—	±0.3	±0.5	
零刻度误差 V_{ADIN} = V_{SSA}	10 位模式	P	E_{ZS}	—	±0.5	±1.5	LSB[2]
	8 位模式			—	±0.5	±0.5	
满刻度误差 V_{ADIN} = V_{DDA}	10 位模式	P	E_{FS}	—	±0.5	±1.5	LSB[2]
	8 位模式			—	±0.5	±0.5	
量化误差	10 位模式	D	E_Q	—	—	±0.5	LSB[2]
	8 位模式			—	—	±0.5	

表 A-9. 5 V 10 位 ADC 特性 (V_{REFH} = V_{DDAD}, V_{REFL} = V_{SSAD})

特性	条件	C	条件	符号	最小值	典型值[1]	最大值
输入漏电流误差 板漏电流[3] * R_{AS}	10 位模式	D	E_{IL}	—	±0.2	±2.5	LSB[2]
	8 位模式			—	±0.1	±1	

[1] 除非另有说明，典型值是在 V_{DDAD} = 5.0 V, Temp = 25°C, f_{ADCK} = 1.0MH 时测试得到。典型值仅作参考用，在生产中未测试。

[2] 1 LSB = $(V_{REFH} – V_{REFL})/2^N$

[3] 基于输入板漏电流。请参考板电气特性。

A.9 内部时钟发生模块特性

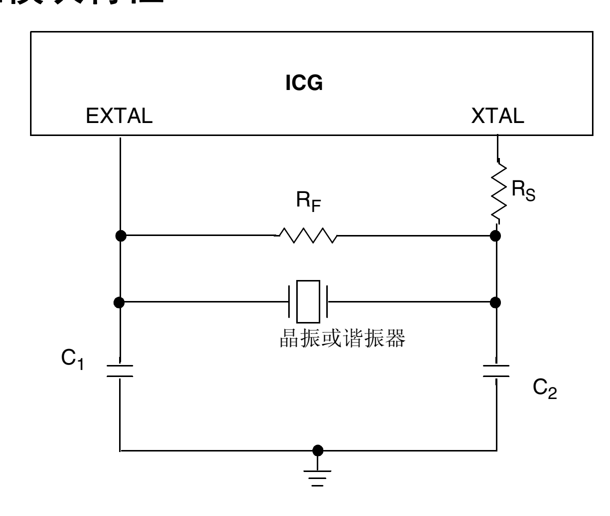


表 A-10. ICG DC 电气规范 (温度范围 = –40 to 125°C 环境温度)

特性	符号	最小值	典型值[1]	最大值	单位
载荷电容器	C_1 C_2	见注 [2]			
反馈电阻器 低量程 (32 ～ 100 kHz) 高量程 (1 ～ 16 MHz)	R_F		10 1	—	MΩ
串行电阻器 低量程 低增益 (HGO = 0) 高增益 (HGO = 1) 高量程 低增益 (HGO = 0) 高增益 (HGO = 1) ≥ 8 MHz 4 MHz 1 MHz	R_S	 — — — — — —	 0 100 0 0 10 20	 — — — — — —	kΩ

[1] 除非另有说明，典型值是温度在 25° C 时的特性表征数据 。

[2] 参见晶振或谐振器生产商的建议。

A.9.1 ICG 频率规范

表 A-11. ICG 频率规范

(V_{DDA} = V_{DDA} 最小值到最大值 , 温度范围 = –40 to 125°C 周围温度)

编号	C	特性	符号	最小值	典型值[1]	最大值	单位
1	T	晶体振荡器或谐振器 (REFS = 1) (基础模式晶体或陶磁谐振器) 低量程 高量程 高增益 , FBE (HGO = 1,CLKS = 10) 高增益 , FEE (HGO = 1,CLKS = 11) 低功率 , FBE (HGO = 0, CLKS = 10) 低功率 , FEE (HGO = 0, CLKS = 11)	f_{lo} f_{hi_byp} f_{hi_eng} f_{lp_byp} f_{lp_eng}	32 1 2 1 2	— — —	100 16 10 8 8	kHz MHz MHz MHz MHz
2	T	输入时钟频率 (CLKS = 11, REFS = 0) 低量程 高量程	f_{lo} f_{hi_eng}	32 2	— —	100 10	kHz MHz
3	T	输入时钟频率 (CLKS = 10, REFS = 0)	f_{Extal}	0	—	40	MHz
4	T	内部参考频率 (未调整)	$f_{ICGIRCLK}$	182.25	243	303.75	kHz
5	T	输入时钟的负载周期 (REFS = 0)	t_{dc}	40	—	60	%
6	P	输出时钟 ICGOUT 频率 CLKS = 10, REFS = 0 所有其它情况	f_{ICGOUT}	f_{Extal} (min) f_{lo} (min)	— —	f_{Extal} (max) $f_{ICGDCLKmax}$(max)	MHz
7	T	最小 DCO 时钟 (ICGDCLK) 频率	$f_{ICGDCLKmin}$	8	—		MHz
8	T	最大 DCO 时钟 (ICGDCLK) 频率	$f_{ICGDCLKmax}$		—	40	MHz
9	P	自时钟模式 (ICGOUT) 频率 [2]	f_{Self}	$f_{ICGDCLKmin}$		$f_{ICGDCLKmax}$	MHz
10	T	自时钟模式复位 (ICGOUT) 频率	f_{Self_reset}	5.5	8	10.5	MHz
11	T	参考频率丢失 [3] 低量程 高量程	f_{LOR}	5 50		25 500	kHz
12	T	DCO 频率丢失 [4]	f_{LOD}	0.5		1.5	MHz
13	T	晶振启动时间 [5,6] 低量程 高量程	t_{CSTL} t_{CSTH}	— —	430 4	— —	ms
14	T	FLL 锁定时间 , [7] 低量程 高量程	t_{Lockl} t_{Lockh}	— —		2 2	ms
15	T	FLL 频率解锁范围	n_{Unlock}	–4*N		4*N	计数
16	T	FLL 频率锁闭范围	n_{Lock}	–2*N		2*N	计数
17	T	ICGOUT 周期抖动，[8] 在 f_{ICGOUT} Max 时测量 长期抖动 (平均 2 ms 的间隔值)	C_{Jitter}	—		0.2	% f_{ICG}
18	P	内部振荡器与调整后 频率[9] 的偏移 V_{DD} = 2.7 – 5.5 V, (恒定温度) V_{DD} = 5.0 V ±10%, –40° C to 125°C	ACC_{int}	— —	±0.5 ±0.5	±2 ±2	%

[1] 除非另有说明，典型值是温度在 25° C 时的特性表征数据 。

[2] 自时钟频率指 FLL 为开路时 DCO 生成的频率。

[3] 参考频率丢失指内部探测到的参考 频率，如果它没有在目标范围内，则将 ICG 转为自时钟模式。

4 DCO 频率丢失指内部探测到的 DCO 频率，如果它没有在目标范围内，则将 ICG 转为旁路 FLL 的外部模式（如果外部参考时钟存在的话）。

5 这个参考是鉴定前的特性表征值，不是 100% 测试值。

6 要获得规范值必须遵守正确的 PC 板布局流程。

7 这个规范适用于 FLL 进入要处的内部或外部模式后进行锁定所需要的时间周期。如果使用晶振 / 谐振器作为参考源，则该规范假定它已经运行。

8 抖动是当 f_{ICGOUT} 最大时在指定间隔内测量的与编程频率的平均偏差。测量采用由已过滤的电源供电的器件，并由稳定的外部时钟信号给出时间。经由 V_{DDA} 和 V_{SSA} 引入的噪音及晶体振荡器频率偏移会提高给定间隔的 C_{Jitter} 比例。

9 参见 图 A-9。

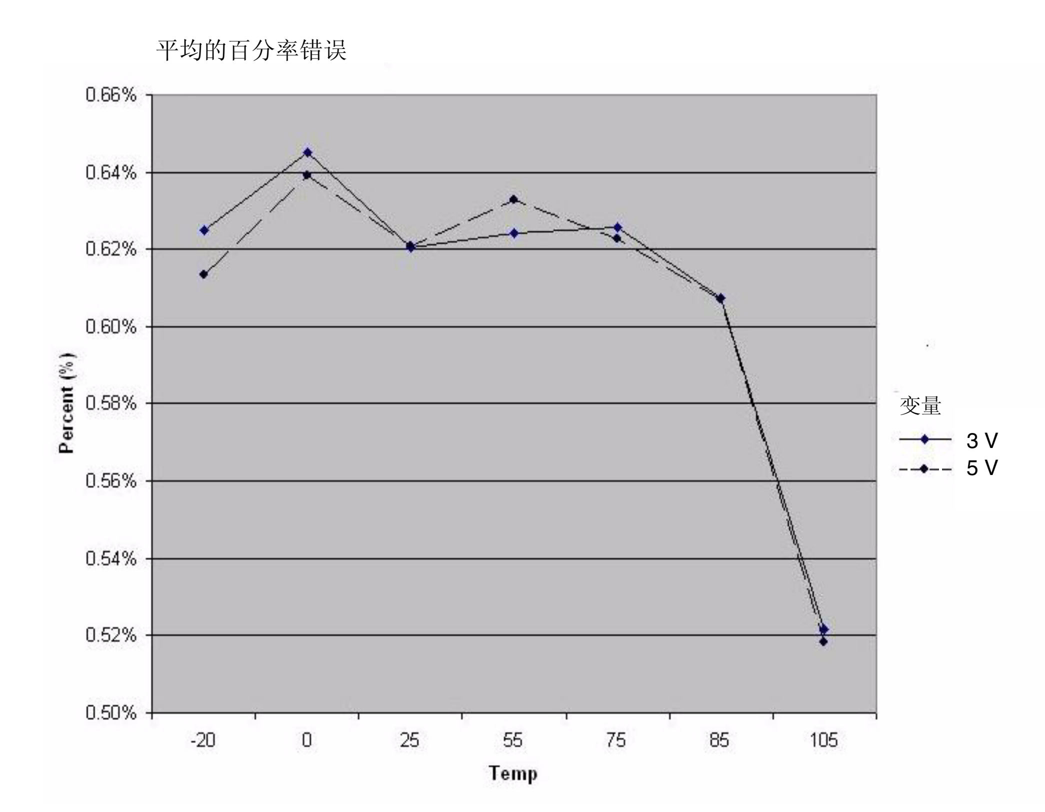


图 A-9. 内部振荡器与调整后频率的偏移

A.10 AC 特性

本小节描述每个外设系统的 AC 定时特性。有关如何生成总线时钟的详细信息，请参见 第 10 章 内部时钟发生器 (S08ICGV4)。

A.10.1 控制时序

表 A-12. 控制时序

编号	C	参数	符号	最小值	典型值[1]	最大值	单位
1		总线频率 (t_{cyc} = 1/f_{Bus})	f_{Bus}	dc	—	20	MHz
2		实时中断内部振荡器周期	t_{RTI}	700		1300	μs
3		外部复位脉冲宽度[2] (t_{cyc} = 1/f_{Self_reset})	t_{extrst}	1.5 x t_{Self_reset}		—	ns
4		复位低驱[3]	t_{rstdrv}	34 x t_{cyc}		—	ns
5		激活后台调试模式锁定建立时间	t_{MSSU}	25		—	ns
6		激活后台调试模式锁定保持时间	t_{MSH}	25		—	ns
7		IRQ 脉宽 异步路径 [2] 同步路径[4]	t_{ILIH}, t_{IHIL}	100 1.5 x t_{cyc}	—	—	ns
8		KBIPx 脉宽 异步路径 [2] 同步路径 [3]	t_{ILIH}, t_{IHIL}	100 1.5 x t_{cyc}	—	—	ns
9		端口升降时间 — 高输出驱动 (PTxDS) (载荷 = 50 pF)[5] 斜率控制关闭 (PTxSE = 0) 斜率控制使能 (PTxSE = 1)	t_{Rise}, t_{Fall}	— —	3 30		ns

1 除非另有说明，典型值是温度在 V_{DD} = 5.0 V，25° C 时的特性表征数据 。
2 这是保证能识别为复位管脚请求的最短脉冲。更短脉冲不能保证无效来自内部源的复位请求。
3 当发起任意复位时，内部电路驱动复位引脚的电平低大约 34 个总线周期，然后再对复位引脚采样约 38 个总线周期，将外部复位请求与内部请求区别开来。
4 这是能保证通过引脚同步电路的最小脉宽。更短的脉宽可能不能被识别。在停止模式中，同步被绕过，这样可以识别更短的脉宽。
5 显示的是 20% V_{DD} 和 80% V_{DD} 水平的时序。温度范围 –40°C 至 125°C.

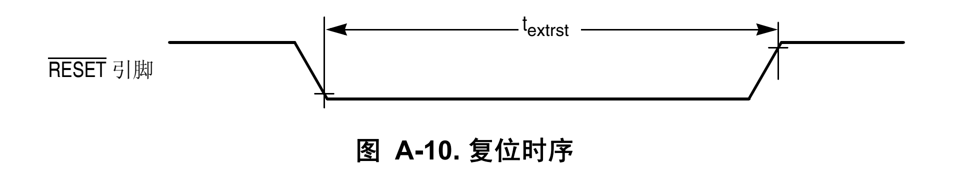


图 A-10. 复位时序

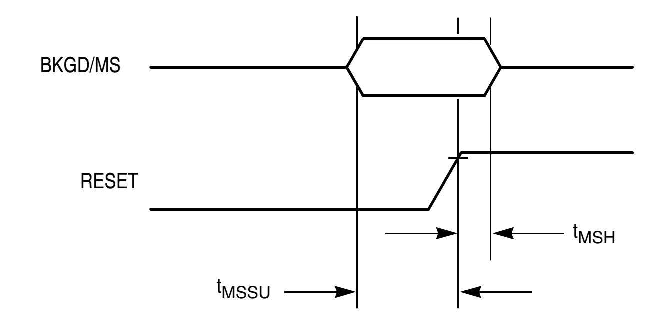


图 A-11. 激活后台调试模式锁定时序

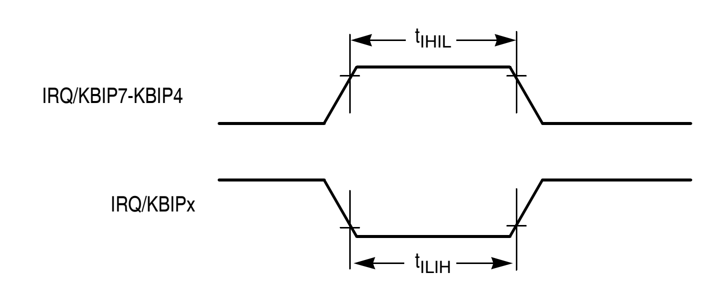


图 A-12. IRQ/KBIPx 时序

A.10.2 定时器 /PWM (TPM) 模块时钟

同步器电路确定可被识别的最短输入脉冲或可以用作定时器计数器的可选外部源的最快时钟。这些同步器从当前的总线速率时钟中操作。

表 A-13. TPM 输入时序

参数	符号	最小值	最大值	单位
外部时钟频率	f_{TPMext}	dc	$f_{Bus}/4$	MHz
外部时钟周期	t_{TPMext}	4	—	t_{cyc}
外部时钟高位时间	t_{clkh}	1.5	—	t_{cyc}
外部时钟低位时间	t_{clkl}	1.5	—	t_{cyc}
输入捕捉脉宽	t_{ICPW}	1.5	—	t_{cyc}

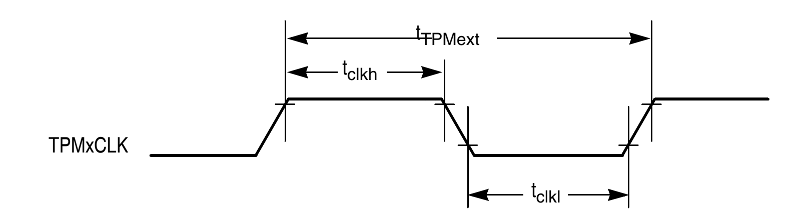


图 A-13. 定时器外部时钟

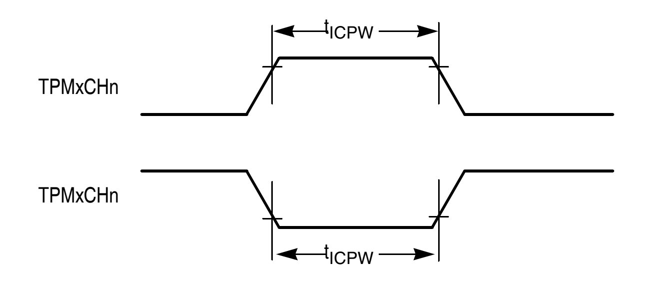


图 A-14. 定时器输入捕捉脉冲

A.11 SPI 特性

表 A-14 和 图 A-15 至图 A-18 描述了 SPI 系统的时序要求。

表 A-14. SPI 电气特性

编号[1]	C	参数[2]	符号	最小值	最大值	单位
		操作频率[3] 主 从	f_{op} f_{op}	$f_{Bus}/2048$ dc	$f_{Bus}/2$ $f_{Bus}/4$	Hz
1		循环时间 主 从	t_{SCK} t_{SCK}	2 4	2048 —	t_{cyc} t_{cyc}
2		使能前置时间 主 从	t_{Lead} t_{Lead}	— 1/2	1/2 —	t_{SCK} t_{SCK}
3		使能时延 主 从	t_{Lag} t_{Lag}	— 1/2	1/2 —	t_{SCK} t_{SCK}
4		时钟 (SPSCK) 高位时间 主从	t_{SCKH}	1/2 t_{SCK} – 25	—	ns
5		时钟 (SPSCK) 低位时间 主从	t_{SCKL}	1/2 t_{SCK} – 25	—	ns
6		数据建立时间 （输入） 主 从	$t_{SI(M)}$ $t_{SI(S)}$	30 30	— —	ns ns
7		数据保持时间 （输入） 主 从	$t_{HI(M)}$ $t_{HI(S)}$	30 30	— —	ns ns
8		访问时间，从[4]	t_A	0	40	ns
9		禁止时间，从[5]	t_{dis}	—	40	ns
10		数据建立时间 （输出） 主 从	t_{SO} t_{SO}	25 25	— —	ns ns
11		数据保持时间 （输出） 主 从	t_{HO} t_{HO}	–10 –10	— —	ns ns

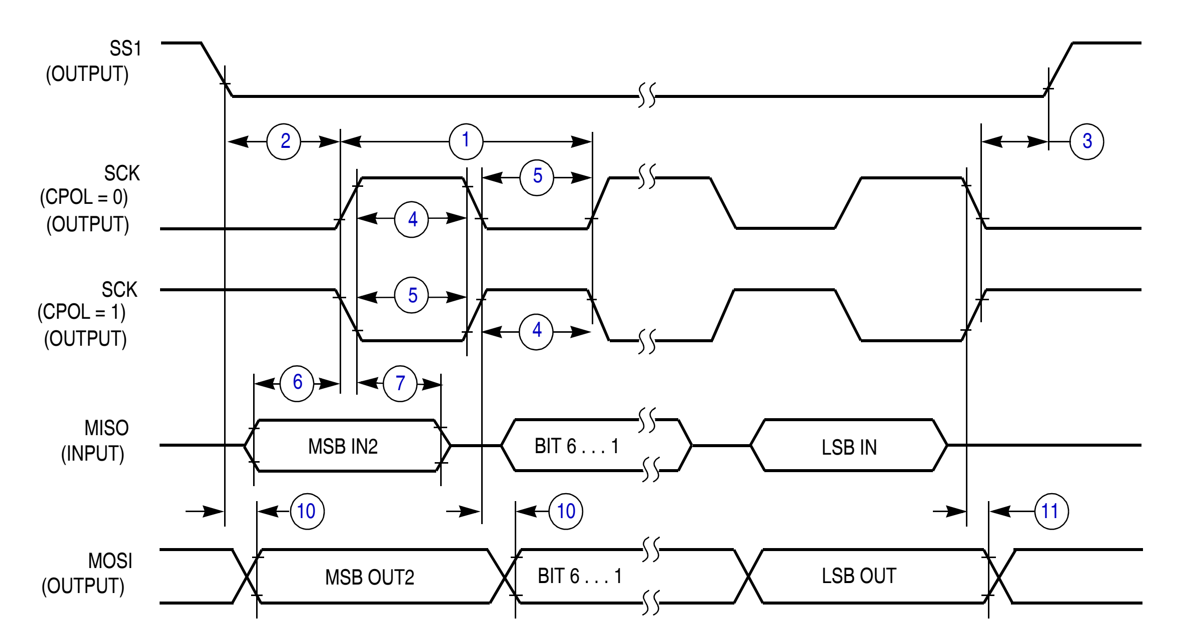
1 参见图 A-15 至图 A-18.

2 除非另有说明，所显示的均为 20% V_{DD} 和 70% V_{DD} 的时钟。所有 SPI 引脚上为 100 pF 载荷。所有时钟假设 SPI 输出管脚禁用斜率控制，启用高驱强度。

3 因为引脚输入输入特性，最大波特率必须限制在 5 MHz 。

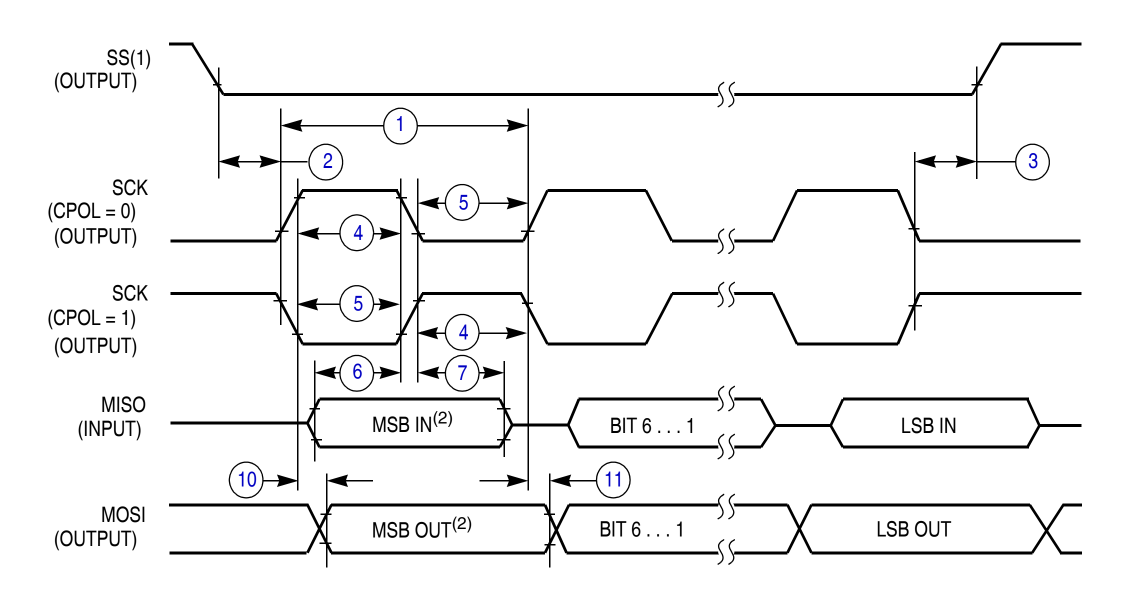
4 从高阻抗状态到数据激活的时间。

5 到高阻抗状态的保持时间。



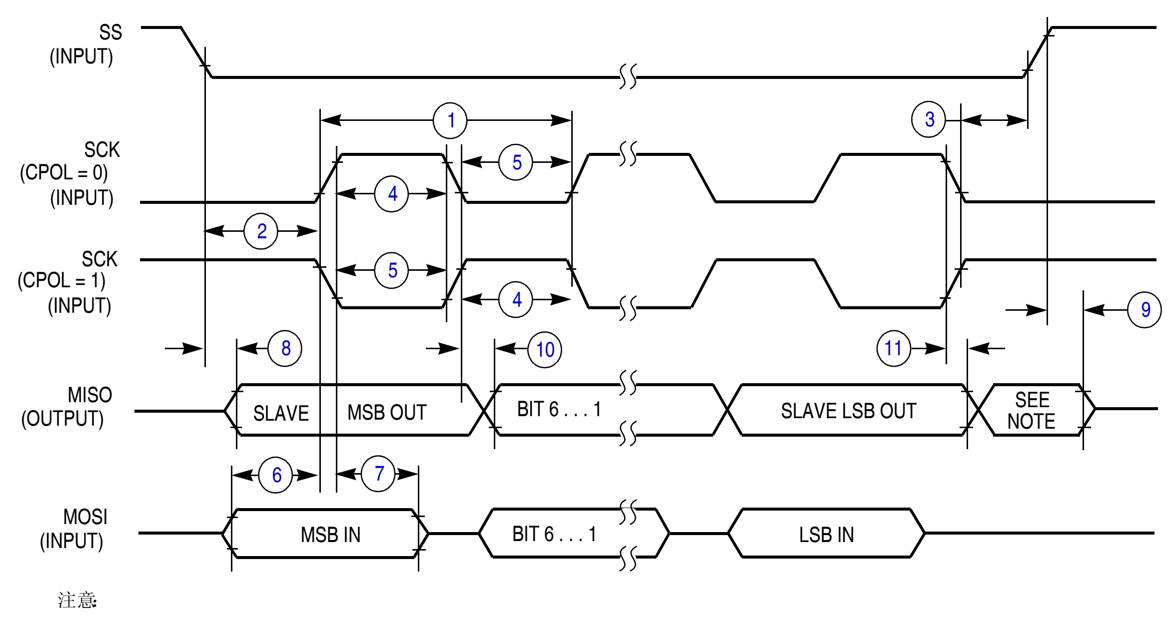
注意：
1. $\overline{SS}$ 输出模式 (MODFEN = 1, SSOE = 1).
2. LSBF = 0。对于 LSBF = 1, 位顺序是 LSB, bit 1, ..., bit 6, MSB.

图 A-15. SPI 主时序 (CPHA = 0)



Þ¢“,£½
1. $\overline{SS}$ 输出模式 (MODFEN = 1, SSOE = 1).
2. LSBF = 0。对于 LSBF = 1, 位顺序为 LSB, bit 1, ..., bit 6, MSB.

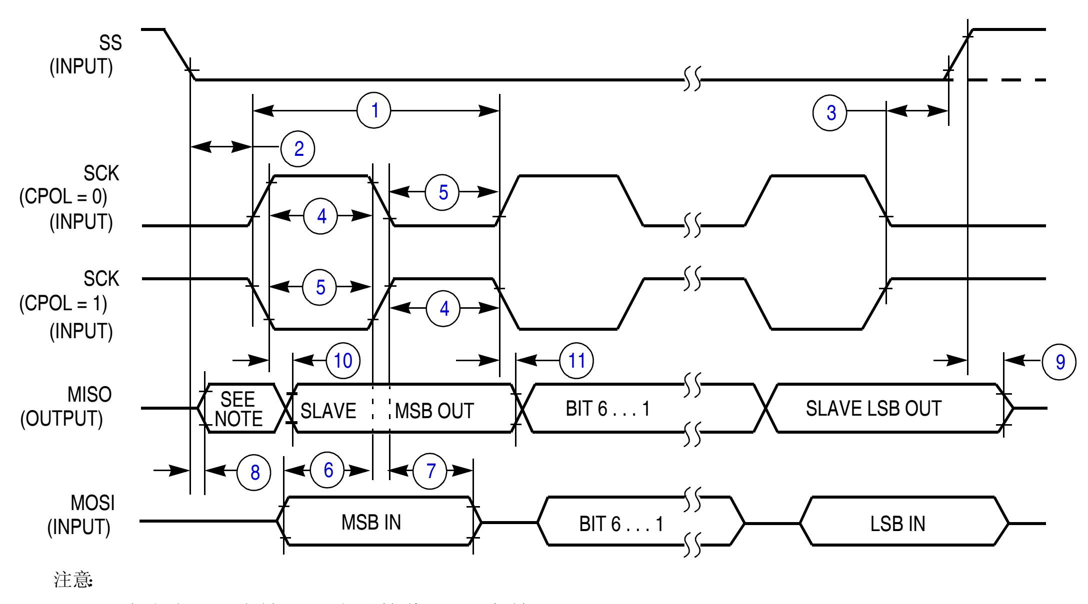
图 A-16. SPI 主时序 (CPHA = 1)



注意:

1. 未定义，但是在正常情况下为刚接收 MSB 字符

图 A-17. SPI 从时序 (CPHA = 0)



注意:

1. 未定义，正常情况下为刚接收 LSB 字符。

图 A-18. SPI 从时序 (CPHA = 1)

A.12 Flash 规范

本小节详细地描述了 FLASH 的编程 / 擦除次数及编程 - 擦除容限。

编程和擦除操作除正常 V_{DD} 电源外不需要任何特殊电源。有关编程 / 擦除操作的更多信息，请参见 µ⁄4’¬ ¥Ê¥¢ý˜。

表 A-15. Flash 特性

编号	C	特性	符号	最小值	典型值[1]	最大值	单位
1	P	编程 / 擦除的电源电压	$V_{prog/erase}$	2.7		5.5	V
2	P	读取操作的电源电压	V_{Read}	2.7		5.5	V
3	P	内部 FCLK 频率[2]	f_{FCLK}	150		200	kHz
4	P	内部 FCLK 周期 (1/FCLK)	t_{Fcyc}	5		6.67	μs
5	P	字节编程时间 （任意位置）(2)	t_{prog}	9			t_{Fcyc}
6	C	字节编程时间 （突发模式）(2)	t_{Burst}	4			t_{Fcyc}
7	P	页面擦除时间[3]	t_{Page}	4000			t_{Fcyc}
8	P	全部擦除时间 (2)	t_{Mass}	20,000			t_{Fcyc}
9	C	编程 / 擦除容限[4] T_L 至 T_H = –40°C 至 125°C T = 25°C		10,000 —	— 100,000	— —	cyces
10	C	数据保留时间[5]	t_{D_ret}	15	100	—	years

1 除非另有说明，所显示的均为 V_{DD}= 5.0 V，25° C 时的特性表征数据。

2 该时钟的频率可以通过软件设置来控制。

3 这些值是硬件状态机控制的值。用户代码无需计算周期数。提供该信息是为了计算编程和擦除的大约时间。

4 **FLASH 的典型容限**，是在 9S12Dx64 上对该产品系列进行评估得到的。有关飞思卡尔半导体如何定义典型容限的更多信息，请参考 Engineering Bulletin EB619/D，*非易失性存储器的典型容限*。

5 典型数据保留时间值基于测试技术的内在能力，它在高温，并使用阿伦尼乌斯公式降到 25°C 时测量。有关飞思卡尔半导体如何定义典型数据保留时间的更多信息，请参考 Engineering Bulletin EB618/D，*非易失性存储器的典型数据保留时间*。

A.13 EMC 性能

电磁兼容性 （EMC）在很大程度上取决于 MCU 所在的环境。电路板设计及布局、电路拓扑选择、外部组件的位置和特性以及 MCU 软件操作等都在 EMC 性能中起到重要作用。系统设计人员应参照飞思卡尔应用笔记，如 AN2321、 AN1050、AN1263、 AN2764 和 AN1259，获取优化 EMC 性能的建议和指导。

A.13.1 瞬态传导抗扰度

微控制器瞬态传导抗扰度根据飞思卡尔内部测试方法测量。测量时，微控制器安装在定制的 EMC 评估板上，同时运行与测试方法相符的专用 EMC 测试软件。在微处理器的每个引脚上注入瞬态传导抗扰度测试信号来确定传导抗扰度。瞬态波形和注入方法参考了 IEC 61000-4-4 (EFT/B)。测试配置中造成任何引脚性能下降所需要的瞬态电压大于或等于报告的水平，否则下表的注脚将另行指出。

表 A-16. 瞬态传导抗扰度

参数	符 号	条件	f_{OSC}/f_{BUS}	结果	振幅[1] (最小值)	单位
传导抗扰度，电快速瞬态 / 突发 (EFT/B)	V_{CS_EFT}	V_{DD} = 5.0V T_A = +25^oC 封装类型 64 QFP	32.768 kHz 晶振 2 MHz 总线	A	±2.8[2]	kV
				B	±2.8	
				C	±2.8	
				D	±3.8	

1 数据基于质量测试结果，未在生产中测试。

2 RESET 引脚对应用的最低 220 V 瞬态电压敏感。但是 ，增加建议的 0.1μF 去偶电容后，应该可以防止低于最低振幅的失败。

表 A-17 给出了抗扰度性能的分类。

表 A-17. 抗扰度性能的分类

结果	性能标准	
A	无故障	干扰时，干扰后， MCU 可按设计标准运行 。
B	自恢复故障	干扰时， MCU 不能按设计标准运行 。干扰消除后， MCU 可自动恢复到正常状态。
C	软故障	干扰时， MCU 不能按设计标准运行 。在干扰消除，且 RESET 引脚复位后， MCU 才可恢复到正常状态。 .
D	硬故障	在出现干扰时， MCU 不能按设计标准运行 。在干扰消除，且对 MCU 执行断电和加电操作后， MCU 才可恢复到正常状态。
E	损坏	干扰时和干扰后， MCU 不能按设计标准运行 。因物理损坏或其它永久性能下降， MCU 不能恢复到正常状态。

附录 B
订购信息和机械图

B.1 订购信息

本小章介绍了 MC9S08AC60 系列器件的订购信息。

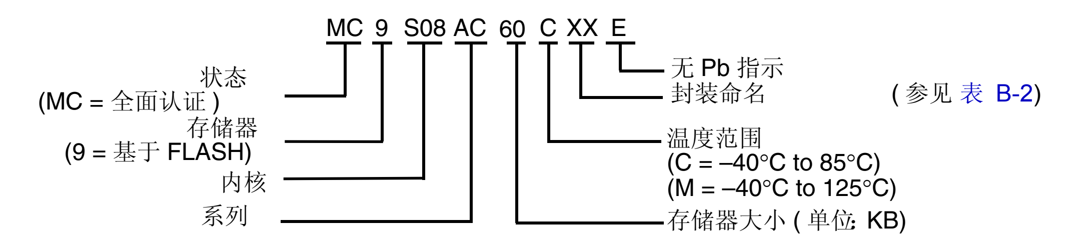
器件编号体系举例如下：

表 B-1. 器件编号体系

器件编号[1]	存储器		可提供封装[2]
	Flash	**RAM**	**类型**
MC9S08AC60	63,280	2048	64 LQFP, 64 QFP 48 QFN, 44 LQFP, 32 LQFP
MC9S08AC48	49,152		
MC9S08AC32	32,768		

[1] 参见表 1-1，了解每个器件上包含的模块。
[2] 参见 表 B-2，了解封装信息。

B.2 可订购部件编号体系



B.3 机械图

以下描述 MC9S08AC60 系列不同封装的机械规范 。参见表 B-2，了解每种封装对应的文档编号。

表 B-2. 封装信息

引脚 数	类型	命名	文档编号
64	LQFP	PU	98ASS23234W
64	QFP	FU	98ASB42844B
48	QFN	FD	98ARH99048A
44	LQFP	FG	98ASS23225W
32	LQFP	FJ	98ASH70029A

如何联系我们：

主页：
www.freescale.com

技术支持网页：
http://www.freescale.com/support

美国 / 欧洲或未列出的地方：
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

欧洲、中东和非洲：
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

日本：
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

亚太地区：
飞思卡尔半导体（中国）有限公司 100022
北京市朝阳区建国路乙 118 号
京汇大厦 23 层
+86 10 5879 8000
support.asia@freescale.com

仅提供印刷品请求：
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

文档号：MC9S08AC60ZHS
第 2 版
2008 年 3 月